



BENEMÉRITA
UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA
COMPUTACIÓN

Sistema de evaluación de las actividades de un
curso

Tesis que para obtener el título de
INGENIERO EN CIENCIAS DE LA
COMPUTACIÓN

PRESENTA:
MARTÍN ROLDÁN ARAUJO

Asesor: M.C. PEDRO BELLO LOPEZ

Julio, 2016

Agradecimientos

A mi familia, compañeros, profesores, a la comunidad de tecnologías de la información de Puebla y a mi asesor de tesis, gracias Pedro.

Contenido

Introducción	6
Objetivos Generales y Específicos del Proyecto.	6
Objetivo general.....	6
Objetivos Específicos.....	6
Capítulo I. Marco de Referencia.....	8
1.1 La ingeniería de software.	8
1.1.1 Objetivos de la ingeniería de software	8
1.2 Metodología de Desarrollo Basado en: Rapid Application Development (RAD)	9
1.2.1 Ciclo de vida del RAD.....	10
1.3 Base de datos	10
1.3.1 Sistema de Gestión de Bases de Datos (SGBD)	10
1.4 Arquitectura Cliente-Servidor	12
1.4.1 Lado del Cliente.....	12
1.4.2 Lado del Servidor.....	12
1.4.3 Programación por Capas	13
1.5 Herramientas de Software para el Desarrollo del Sistema	14
1.5.1 PHP	14
1.5.2 MySQL	15
1.5.3 Apache.....	15
1.5.4 Elementos para la interfaz	16
Capítulo 2. Análisis de requerimientos y Diseño.....	18
2.1 Planteamiento del problema	18
2.1.1 Justificación	18
2.1.2 Objetivo general.....	19
Objetivos específicos.....	19
2.2 Restricciones	19
2.3 Alcance del proyecto.....	20
2.4 Impacto Socioeconómico.	21
2.5 Aportaciones.	21

2.6 Requerimientos	21
2.6.2 Requerimientos de negocio	22
2.6.3 Requerimientos de usuario	22
2.6.4 Requerimientos funcionales	22
2.6.5 Requerimientos no funcionales	23
2.7 Diseño.....	24
2.7.1 Diagramas de casos de uso	24
2.7.2 Clases de Usuario y sus Características	30
2.8 Diseño de la Base de Datos	30
2.8.1 Modelo conceptual de la base de datos	30
2.8.2 Diagrama entidad-relación.....	35
2.8.3 Diagrama de contexto	36
2.9 Diagrama de secuencia	37
2.10 Diagrama de clases.....	38
2.11 Métodos de las clases	40
2.13 Detalle de métodos de clases	41
2.14 Interfaces	42
Capítulo 3. Construcción e Implementación.....	49
3.1 Configuración del Servidor	49
3.1.1 Infraestructura	49
3.1.2 El Sistema Operativo: Ubuntu 12.04 LTS.....	49
3.1.3 Apache y PHP	49
3.2 Acceso al servidor	49
3.3 Configuración del nombre de dominio y DNS.....	50
3.4 Localización de archivos del proyecto	50
3.5 Permisos de directorios y archivos	50
3.6 MySQL	50
3.7 Usuario para la Base de datos del Sistema	50
3.7.1 PhpMyAdmin.....	51
3.8 Firewall	51

3.9 PHP: Yii Framework MVC	51
3.9.1 Módulo Cruge User	52
3.9.2 JavaScript y HTML5: Framework Kendo UI	52
Capítulo 4. Uso	54
4.1 Uso del Producto	54
4.2 Disponibilidad.....	54
4.3 Arranque y Carga Inicial	54
4.4 Características Generales del Producto	54
4.5 Funcionamiento Global del Producto	54
4.6 Supuestos Generales.....	54
4.7 Riesgos y Dependencias	55
4.8 Calidad de datos.....	55
4.9 Reglas Generales de esta Aplicación.....	55
4.9.1 De Captura y Navegación	55
4.9.2 De Comportamiento	55
4.9.3 Mensajes de Confirmación / Advertencia y de Error	56
Conclusiones	57
Mejoras a futuro	57
Bibliografía.	58

Introducción

Actualmente, los profesores llevan registros de las actividades de los alumnos en hojas, libretas bitácoras y demás elementos que se deterioran, pierden o son falsificables dada la naturaleza de los materiales con los que están contruidos.

En otros casos, se lleva el registro en hojas de cálculo electrónicas (como Microsoft Excel) pero ésta forma de llevar el registro tiene algunas desventajas:

- Sólo el profesor tiene acceso a la lista.
- Los alumnos no pueden ver su registro en tiempo real.
- No se puede ver vía internet.
- Vulnerable a pérdida de información o incluso pérdida del archivo.
- En su mayoría, pago de licencia por computadora que se esté usando.
- No se puede visualizar en muchos dispositivos móviles.
- El equipo debe tener un software de hoja de cálculo compatible para ver el archivo.
- La gestión de las actividades, si se tienen muchos cursos, alumnos y rúbricas de evaluación, puede llegar a ser complicada.

Es por eso, que se propone el desarrollo de este sistema, para ayudar al profesor en la gestión de las actividades de los alumnos sin preocuparse por los puntos mencionados anteriormente.

Objetivos Generales y Específicos del Proyecto.

A continuación se presentan los objetivos generales y específicos del proyecto.

Objetivo general

Desarrollar un sistema en web que permita llevar el control de las actividades de un grupo de una determinada asignatura con respecto a los exámenes, proyecto, tareas, asistencia y demás actividades.

Objetivos Específicos

El sistema permitirá capturar qué elementos se evaluarán en un curso considerando los más genéricos como son exámenes, tareas, asistencia, proyecto final, prácticas de laboratorio, etc., además de cada uno de estos elementos se podrán tener uno o más extras.

El sistema tendrá que administrar más de un curso y al final deberá obtener los promedios de los alumnos inscritos en el curso de acuerdo a los porcentajes establecidos para cada rubro.

Además se debe desarrollar las siguientes funcionalidades:

- Diseñar la base de datos relacional
- Crear una interfaz amigable para la administración de la información
- Generar consultas
- Publicar resultados parciales y finales

Este documento está compuesto por cuatro Capítulos:

Capítulo I: Definiciones, conceptos y explicaciones que son necesarias para el lector y éste entre en contexto con el entorno de desarrollo del sistema. Además se listan y describen las herramientas usadas para el mismo.

Capítulo II: Análisis de requerimientos y Diseño. En este capítulo se analizan los requerimientos del sistema y se modelan, se desarrolla el diseño conceptual, que establece la primera fase de desarrollo de base de datos y basándose en el esquema obtenido se desarrolla el diseño lógico de la base de datos. En este capítulo definirán entidades, atributos y relaciones mediante un diccionario de datos, datos que se representaran gráficamente mediante el modelo de datos entidad relación, que ofrece una manera de identificar entidades que se van a representar en la base de datos y el modo en que se relacionan entre sí. Una vez obtenido el modelo entidad relación, se desarrollara el modelo relacional el cual ya depende del SGBD de MySQL, Apache y PHP del lado del servidor.

Capítulo III: Implementación y pruebas. Ya teniendo diseñados los módulos que componen al sistema, tanto la interfaz, bases de datos, relaciones, búsquedas se integra y se realizan pruebas para examinar el desempeño del mismo. Este último supone etapa final del desarrollo y con ello la culminación de la tesis.

Capítulo IV: Conclusiones y Uso del Producto. Se describen las experiencias y pensamientos finales respecto al desarrollo del sistema, así como el uso del producto final.

Capítulo I. Marco de Referencia

En este primer capítulo se describen los diversos temas relacionados con el desarrollo del sistema.

1.1 La ingeniería de software.

El termino ingeniería del software se presentó a finales de los años sesentas, para expresar el área de conocimientos que se desarrollan en torno a un proyecto y las problemáticas que se ofrecían en el software.

Se le llama ingeniería de software, a la disciplina obtenida por diferentes tipos de técnicas para la elaboración y el manejo de proyectos de programación, también llamada ingeniería informática.

“La ingeniería del software es una disciplina de la ingeniería cuya meta es el desarrollo costeable de sistemas de software. Éste es abstracto e intangible” [6].

“Es el área de la ingeniería que ofrece métodos y técnicas para desarrollar y mantener software” [8]

1.1.1 Objetivos de la ingeniería de software

En la arquitectura y proceso de proyectos se utilizan procedimientos y técnicas para resolver los problemas, la informática contribuye a herramientas y procedimientos en la cual se crea la ingeniería de software.

- Mejorar la calidad de los productos de software
- Aumentar la productividad y trabajo de los ingenieros del software.
- Facilitar el control del proceso de desarrollo de software.
- Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
- Definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado.

1.2 Metodología de Desarrollo Basado en: Rapid Application Development (RAD)

“El desarrollo rápido de aplicaciones o RAD (acrónimo en inglés de rapid application development) es un proceso de desarrollo de software, desarrollado inicialmente por James Maslow en 1980. El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución” [13].

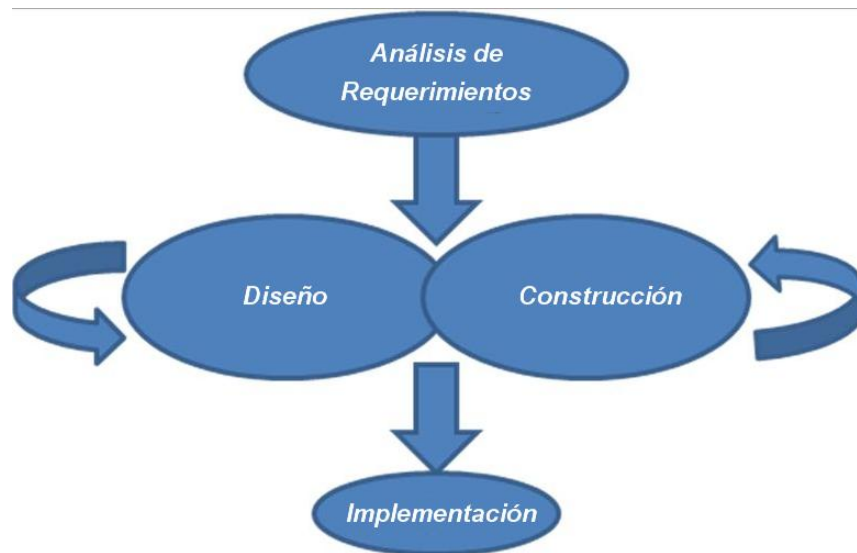


Ilustración 1.1 Modelo RAD

Se opta por ésta metodología ya que se recomienda para proyectos con periodo de desarrollo entre 60 a 90 días. Está diseñado para crear aplicaciones de alta calidad.

Por ejemplo, Hace algunos años un proyecto para desarrollar un sistema informático duraba 18 a 24 meses; actualmente, con la utilización del modelo RAD toma entre 1 a 3 meses aproximadamente.

Se muestra la evolución de cada etapa de éste proyecto en los capítulos siguientes.

1.2.1 Ciclo de vida del RAD

Análisis de requerimientos: En ésta etapa se definen las funciones del negocio y el alcance del sistema mediante una estructurada discusión sobre los problemas que necesitan solución.

Diseño: Consiste de un análisis detallado de las actividades requeridas por el cliente en relación a la aplicación propuesta. Una vez completado el análisis, se crean los diagramas que definen las entidades, los procesos y la información. Se traza el diseño del sistema y se desarrollan los procedimientos, así como los esquemas de las interfaces.

Construcción: La construcción de la aplicación reside de una cadena de pasos donde los usuarios tienen la oportunidad de afirmar los requisitos y repasar los resultados (muy importante). Los puntos más fuertes en ésta etapa son la programación y desarrollo del sistema.

Implementación: Incluye la puesta a prueba y la capacitación del usuario final, la conversión de datos y la implementación de la aplicación. Como resultado, el nuevo sistema es construido, entregado y puesto en operación anticipadamente.

1.3 Base de datos

Una Base de Datos es un conjunto de información electrónica ordenada de forma que un programa pueda seleccionar y usar los datos que necesite para el fin que se requiera.

- Tradicionalmente se organizan por campos, registros y archivos.

Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros.

Para tener acceso a la información en una Base de Datos, es necesario un Sistema de Gestión de Bases de datos.

1.3.1 Sistema de Gestión de Bases de Datos (SGBD)

Es una colección de programas que permiten almacenar, modificar y extraer información de una base de datos.

Se especializa en interconectar e intercomunicar la(s) base(s) de datos, los usuarios y los programas que hacen uso de las mismas. Su propósito es el de

manejar de una manera sencilla, ordenada y práctica uno o varios conjuntos de datos que posteriormente serán de utilidad para el usuario.

Existen distintos objetivos que deben cumplir los SGBD:

- **Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Consistencia.** En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea. Por otra parte, la base de datos representa una realidad determinada que tiene determinadas condiciones, por ejemplo que los menores de edad no pueden tener licencia de conducir. El sistema no debería aceptar datos de un conductor menor de edad. En los SGBD existen herramientas que facilitan la programación de este tipo de condiciones.
- **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Manejo de transacciones.** Una transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que luego de una ejecución en la que se produce una falla es el mismo que se obtendría si el programa no se hubiera ejecutado. Los SGBD proveen mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.
- **Tiempo de respuesta.** Lógicamente, es deseable minimizar el tiempo que el SGBD demora en proporcionar la información solicitada y en almacenar los cambios realizados.

Algunos ejemplos son MySQL, PostgreSQL, Oracle SQL, SQL Server, Firebird, entre muchos otros.

1.4 Arquitectura Cliente-Servidor

Es una arquitectura de red en la cual cada ordenador o proceso en la red es un cliente o un servidor. Los Servidores por lo regular son potentes computadoras o procesos dedicados a gestionar unidades de disco (servidores de archivo), impresoras (Servidor de impresiones) o el tráfico de una red (Servidores de red).

Los Clientes son ordenadores o estaciones de trabajo en las cuales los usuarios ejecuten aplicaciones. Dependen de servidores para explotar recursos, como archivos, dispositivos e incluso poder de procesamiento.

En el contexto de este documento, se utiliza esta arquitectura para implementar un Servidor Web y un servidor de bases de datos el cual calcula y brinda la información relevante a la máquina cliente.

Los clientes hacen peticiones para tener acceso a la base de datos del servidor.

1.4.1 Lado del Cliente

Toma sentido en una arquitectura Cliente-Servidor, significa que la ejecución de los programas o scripts se realiza en el navegador del usuario. El navegador web se llama también cliente web y se denomina "cliente" porque hace las tareas de solicitud y consumo de servicios. El navegador o cliente web se conecta con un servidor, al que solicita páginas. El servidor web se las sirve para consumo del cliente, que las muestra a la persona que las ha solicitado.

Los lenguajes del cliente son los que se ejecutan en el cliente web, como Javascript, CSS, HTML, Flash, etcétera.

1.4.2 Lado del Servidor

El servidor se encarga de procesar las peticiones de los clientes para después devolver una respuesta que corresponda y tenga congruencia a las peticiones de las máquinas cliente.

Las operaciones que se realizan del lado del servidor, requieren acceso a información o funcionalidades que no están disponibles del lado del cliente; por ejemplo, una base de datos o un motor intérprete, como por ejemplo PHP o ASP.

1.4.3 Programación por Capas

En la programación en capas los objetos se dividen según su funcionalidad. Destacan tres principales: la Capa de Interfaz o Frontera, compuesta por los objetos encargados de interactuar con el usuario, como lo son los formularios e interfaces de la aplicación; por otra parte está la Capa de Lógica de Negocio o Control, en donde se encuentran los objetos que realizan la mayor parte del trabajo interno del programa, en esta etapa destaca la lógica de la aplicación así como la funcionalidad de servir de enlace entre las otras capas; por último se encuentra la Capa de Datos, integrada por los objetos que envían y obtienen información al comunicarse con bases de datos u otros sistemas de información que colaboran con el programa.

Dentro del concepto de programación en capas, existen dos términos esenciales para el mejor entendimiento de los conceptos relativos a esta metodología, es aquí donde radica la importancia de la cohesión y el acoplamiento dentro de una aplicación generada mediante este método.

- **Cohesión:** Este término es utilizado para describir el comportamiento que deben tener los módulos y objetos de un sistema o subsistema, comportamiento que describe la forma en que deben trabajar los objetos y módulos entre sí, con alta cohesión para que trabajando en conjunto los módulos y objetos puedan alcanzar un solo propósito de manera más eficaz y rápida. Determina que las operaciones de un objeto deben trabajar en conjunto para alcanzar un propósito común. Es deseable que haya alta cohesión.
- **Acoplamiento:** Se refiere al grado de dependencia que existe entre los módulos. Este grado de dependencia debe ser considerablemente bajo ya que el trabajo se divide en módulos para que cada uno tenga un funcionamiento específico y puede ser más factible la implementación por separado de cada uno. En caso de haber alto acoplamiento entre módulos no se estaría alcanzando el principal objetivo de este modelo, el cual es dividir una tarea grande en varias pequeñas, ya que los módulos actuarían como uno solo al estar altamente acoplados entre sí y se perdería el objetivo primordial de dividir el proyecto.
- **Ventajas:** Al implementar éste modelo de programación, se asegura un trabajo de forma ordenada y separada, debido a que sigue el principio de “divide y vencerás”. Cada capa está dividida según su funcionalidad cuando se quiere modificar el sistema basta con

cambiar un objeto o conjunto de objetos de una capa. Esto se llama modularidad.

- **Desventajas:** Cuando se implementa un modelo de programación en capas, se debe llegar a un balance entre el número de capas y subcapas que componen el programa. Este debe ser el necesario y suficiente para realizar un trabajo específico con eficiencia y ser lo más modular posible. De lo contrario se tiene una serie de desventajas como: pérdida de eficiencia, realización de trabajo innecesario o redundante entre capas, gasto de espacio de la aplicación debido a la expansión de las capas, o bien una alta dependencia entre los objetos y capas que contradice el objetivo principal del modelo. Esto permite un desarrollo más ordenado, y a la larga los sistemas serán más fáciles de mantener.

1.5 Herramientas de Software para el Desarrollo del Sistema

Las herramientas primordiales en este sistema son MySQL, PHP y Apache (SGBD, Lenguaje del lado del servidor y servidor web respectivamente). Las demás son herramientas que se trabajan en el área no funcional, tales como la estética visual del sistema y la manera en la cual se muestra la información.

1.5.1 PHP

Es un lenguaje de programación diseñado originalmente para la creación de páginas web dinámicas multiusuario. Es usado principalmente en interpretación del lado del servidor, pero actualmente puede ser utilizado desde cualquier sitio en línea sin necesidad de que pertenezca a una red LAN facilitando la comunicación con la interfaz de grafica usando bibliotecas del sistema.

El parecido que tiene el lenguaje PHP con los lenguajes mas familiares de programación estructurada, como C, permiten a los programadores crear aplicaciones complejas con en muy corto tempo ocupando muchas funciones conocidas en los anteriores lenguajes.

Mediante extensiones del lenguaje es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos. También permite la conexión a diferentes tipos de servidores de bases de datos así como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y sqlite.

PHP tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Linux, Mac OS X y Windows, y puede interactuar con los servidores de web más populares Apache, e ISAPI.

PHP es una alternativa a las tecnologías de Microsoft ASP y ASP.NET. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un entorno de desarrollo integrado comercial llamado Zend Studio. Recientemente, la división de lenguajes de programación de Borland, ha sacado al mercado un entorno integrado de desarrollo para PHP, denominado Delphi for PHP.

1.5.2 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Existen varias apis que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi, Java, PHP, etc.... También existe un interfaz ODBC, llamado myodbc que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

“Es rápido, robusto y fácil de usar, además, es el más célebre SGBDR del mundo Open Source” [1].

“Es capaz de administrar bases de datos de gran tamaño. Ciertas empresas lo utilizan actualmente con bases de datos de unas 60000 tablas y alrededor de 5 000 000 000 de registros” [1].

1.5.3 Apache

El servidor HTTP Apache es un software servidor HTTP de código abierto para plataformas Unix, Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, un servidor "emparchado". Apache se desarrolla dentro del proyecto HTTP Server (http) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

1.5.4 Elementos para la interfaz

A continuación se describen los elementos principales para el desarrollo de la interfaz con el usuario.

- **HTML**

Abreviación de HyperText Markup Language, es el lenguaje para crear documentos en la World Wide Web. HTML define la estructura general de un documento Web, usando una variedad de etiquetas y atributos.

Hay cientos de etiquetas que se usan para dar formato a la información en una página web, también se usan para especificar ligas de hipertexto. Estos ayudan a los desarrolladores Web a dirigir a los usuarios a otras páginas Web con sólo un clic del mouse ya sea en imágenes o palabras.

- **JSON**

Abreviación de JavaScript Object Notation, JSON es un formato ligero de intercambio de datos el cual es fácil de leer y escribir para los humanos mientras que para las máquinas parsear y generar.

Está basado en la notación de objetos del lenguaje JavaScript, aún así, no requiere de Javascript para leer o escribir ya que es un formato de texto independiente del lenguaje.

- **Ajax**

Asynchronous JavaScript and XML, es un término que describe un nuevo paradigma que permite usar varias tecnologías juntas, incluyendo: HTML o XHTML, Hojas de Estilo en Cascada, JavaScript, DOM, XML, XSLT, y el objeto XMLHttpRequest.

Cuando estas tecnologías son combinadas en el modelo Ajax, las aplicaciones Web son capaces de hacer rápidas e incrementales actualizaciones a la interfaz de usuario sin necesidad de recargar la página web completa.

- **jQuery**

Es una librería JavaScript libre y de código abierto que es usada por desarrolladores web para navegar por documentos HTML, manejar eventos, animaciones y agregar interacción Ajax a las páginas Web.

“jQuery is built on top of JavaScript, a rich and expressive language in its own right” [7].

- **CSS**

Es una nueva característica agregada a HTML que brinda tanto a usuarios y desarrolladores de sitios Web control sobre el cómo se muestran las páginas.

Con CSS los diseñadores y usuarios pueden crear hojas de estilo que definen cómo se muestran distintos elementos, tales como encabezados, enlaces, imágenes, entre otros.

Dichas hojas de estilo pueden aplicarse a cada página Web.

- **Javascript**

Es un lenguaje de scripting desarrollado por Netscape para permitir a los autores Web diseñar sitios interactivos.

Aunque comparte características con el lenguaje Java, fue desarrollado independientemente.

Javascript puede interactuar con código fuente HTML, permitiendo agregar atractivo contenido dinámico.

Es un lenguaje abierto que cualquiera puede usar sin necesidad de una licencia, lo soportan gran cantidad de navegadores web.

Además se hace uso de herramientas para la edición de código (editores de texto como Notepad++, IDEs como Eclipse), navegación (Internet Explorer, Chrome, Firefox), librerías complementarias propias de PHP (FPDF, PHPMailer) para el correcto desarrollo del Producto.

Capítulo 2. Análisis de requerimientos y Diseño

2.1 Planteamiento del problema

Se desea generar un sistema de gestión de evaluación de las actividades que realizan los alumnos que han sido inscritos a los cursos que imparte cierto profesor, de tal forma que automatice el registro, seguimiento y evaluación de las mismas.

Éste sistema facilitará a los docentes tener mejor control de la evaluación de cada uno de los alumnos, así como llevar de forma más precisa el progreso de ellos en el o los cursos.

El sistema estará disponible a través de internet con el fin de que los alumnos puedan en todo momento saber el avance que llevan en el curso y no esperar hasta el final para saber si tendrán éxito (acreditar) en el curso.

El sistema tendrá que registrar alumnos con su profesor y asignatura correspondiente, se llevará un control de diversas actividades a evaluar. Estas actividades forman parte de un porcentaje, que al final del curso servirá para evaluar a los alumnos.

2.1.1 Justificación

En México, el uso de las tecnologías de la información en la educación no está muy aplicado. Podemos ver, por ejemplo, a profesores de los diversos niveles educativos (desde primaria hasta universidad), llevar el registro de todas las actividades de sus alumnos en simples notas de papel o bitácoras. Esto supone un método no confiable para el control del avance de los alumnos, debido a la naturaleza de los materiales en donde se almacenan.

Por otro lado, las opciones de software que existen actualmente en el mercado son muy generales y con demasiadas opciones de configuración. Estas opciones, si bien pueden cubrir hasta cierto punto dicha necesidad, presentan un problema de practicidad al usuario final debido a los recursos de hardware y software que requieren.

Por eso se propone éste sistema, además de aprovechar el internet para que los alumnos puedan saber en todo momento sus avances en cierto curso.

2.1.2 Objetivo general

Desarrollar un sistema en web que permita llevar el control de las actividades de un grupo de una determinada asignatura con respecto a los exámenes, proyecto, tareas, asistencia y demás actividades que el profesor considere necesarias.

Objetivos específicos

El sistema permitirá al docente capturar qué elementos se evaluarán en un curso considerando los más genéricos como son exámenes, tareas, proyecto final, prácticas de laboratorio. Además de cada uno de estos elementos se podrán tener uno o más extras.

El sistema tendrá que administrar más de un curso y al final deberá obtener los promedios de los alumnos inscritos en el curso de acuerdo a los porcentajes establecidos para cada rubro.

Además se debe desarrollar las siguientes funcionalidades:

- Diseñar la base de datos relacional
- Crear una interfaz amigable para la administración de la información
- Generar consultas
- Publicar resultados parciales y finales

2.2 Restricciones

El sistema contará con las siguientes restricciones:

- La única medida de seguridad es la contraseña del profesor, cifrada a nivel de base de datos.
- Cuenta solamente con una validación básica de los datos (usuario/contraseña).
- Los alumnos no poseen cuenta y/o contraseña para ingresar al sistema, los resultados de las calificaciones y avances en el curso están disponibles al internauta que posea el hipervínculo proporcionado con anterioridad por el profesor.
- El listado de alumnos se obtendrá de un catálogo previamente cargado en la base de datos, emulando así la posible interconexión con otros sistemas.

- El único tipo de seguridad es por medio de contraseña de cada profesor. A cada rubrica de evaluación se debe asignar un porcentaje, creando categorías ellas, el total no debe exceder un total de 100% por curso, conjuntamente con el resto de las rúbricas, ver ejemplo de la tabla 2.1.

Categoría	Porcentaje	Rúbricas	
Tareas	30%	Tarea 1	20%
		Tarea 2	20%
		Tarea 3	20%
		Tarea 4	20%
		Tarea 5	20%
		<i>SubTotal</i>	<i>100%</i>
Exámenes	70%	Examen 1	50%
		Examen 2	50%
		<i>SubTotal</i>	<i>100%</i>
Total del curso	100%		

Tabla 2.1 Ejemplo de actividades y rubricas

2.3 Alcance del proyecto

Más que un sistema LMS (Learning Management System) en su totalidad, lo que se espera es un pequeño sistema que lleve el control de las actividades escolares. No se toma en cuenta el desarrollo colaborativo ni la publicación de contenidos interactivos.

“Learning Management System Technologies and Software Solutions for Online Teaching: Tools and Applications focuses on the interaction between learning management systems and relevant teaching methodologies to bring attention to developments in distance learning technologies and programs”. [4].

Un sistema que permita administrar las evaluaciones de las actividades de los cursos, únicamente con privilegios de editar por el profesor, los alumnos sólo podrán ver (consultar) información a través de internet. Los alumnos no tendrán ningún tipo de privilegio, se entiende como alumno a cualquier internauta interesado en consultar información generada por el sistema.

El sistema estará administrado por los docentes que tendrán que registrar sus cursos y los alumnos de cada curso así como las actividades que desean evaluar.

Los profesores deberán actualizar los avances de los alumnos y podrán eliminar, modificar o agregar más actividades a evaluar.

2.4 Impacto Socioeconómico.

Se busca que este tipo de sistemas se usen más en México en instituciones educativas públicas y privadas. Para que el control por parte del profesor, así como la comunicación con los alumnos, sea de la manera más sencilla, eficiente y transparente.

2.5 Aportaciones.

El manejo de las bases de datos del sistema, consultas, disparadores, procedimientos almacenados; la implementación de un patrón de diseño en un sistema web moderno, la comunicación asíncrona de información mediante una interfaz web con un servidor remoto, así como la administración básica de un servidor Linux, incluyendo la automatización de procesos batch para la notificación a los alumnos por medio de email.

En general, el marco y proceso del desarrollo de un sistema web con tecnologías actuales y la redacción de los documentos de la ingeniería de software servirá de antecedentes en México en el desarrollo de futuros proyectos de naturaleza similar.

2.6 Requerimientos

Los principales actores identificados en el proyecto son:

- Profesor. Es la persona encargada de administrar su propia información, como su nombre, nombre de usuario, contraseña, cursos, rúbricas de evaluación y alumnos de cada curso.
- Alumno. Es la persona inscrita en un curso, interesada en consultar las calificaciones de un profesor por medio de internet.

Los principales requerimientos del sistema, se dividen en Requerimientos de negocio, Requerimientos de usuario, Requerimientos funcionales y Requerimientos no funcionales.

2.6.2 Requerimientos de negocio

- RN01 Contar con un medio para que los profesores lleven el control de las actividades de los alumnos en un curso.
- RN02 Los alumnos pueden revisar sus avances por medio de internet.

2.6.3 Requerimientos de usuario

- RU01 Acceder al sistema por medio de usuario y contraseña
- RU02 Crear, modificar o borrar sus propios cursos.
- RU03 Crear, modificar o borrar alumnos en cada curso.
- RU04 Crear, modificar o borrar rúbricas de evaluación de cada curso.
- RU05 Asignar un porcentaje (de un total de 100%) a un subconjunto de rúbricas.
- RU06 Visualización de la relación de los alumnos de cada curso
- RU07 Publicar el listado de la evaluación de los alumnos de los cursos en internet.
- RU08 Asignar, editar nota en rubrica de evaluación.

2.6.4 Requerimientos funcionales

Para el requisito de usuario RU01 se tienen:

Número	Requerimiento	Descripción
RF01	Detalle de usuario	Permite el ingreso del usuario único de cada profesor
RF02	Detalle de contraseña	Permite el ingreso de una contraseña

Para el requisito de usuario RU02 se tienen:

Número	Requerimiento	Descripción
RF03	Especificar el detalle del curso	El sistema deberá permitir crear o modificar el de NRC y nombre del curso
RF04	Eliminar curso	Permite el borrar el curso, eso implica los alumnos y sus evaluaciones.

Para el requisito de usuario RU03 se tienen:

Número	Requerimiento	Descripción
RF04	Especificación de datos de alumno	Permite el ingreso de datos personales de cada alumno
RF05	Eliminar alumno	Permite el borrar a un alumno y con ello mismo sus evaluaciones.

Para el requisito de usuario RU05 se tiene:

Número	Requerimiento	Descripción
RF06	Especificar un porcentaje a un subconjunto de rúbricas	Permite asignar un porcentaje a rúbricas de evaluación de un total de 100%

Para el requisito de usuario RU06 se tiene:

Número	Requerimiento	Descripción
RF07	Detalle de los alumnos de un curso	Permite ver el listado detalle de los alumnos en un determinado curso

Para el requisito de usuario RU07 se tiene:

Número	Requerimiento	Descripción
RF08	Visualizar el RF07 en internet	El Requerimiento RF07 debe estar accesible por medio de internet.

Para el requisito de usuario RU08 se tiene:

Número	Requerimiento	Descripción
RF09	Gestionar nota en rubrica de evaluación	El profesor puede asignar o editar una nota en alguna rubrica de evolución

2.6.5 Requerimientos no funcionales

Número	Requerimiento	Descripción
RNF01	Usabilidad	Debe ser fácil de usar. Con interfaces intuitivas

RNF02	Seguridad	El ingreso al sistema estará restringido bajo contraseñas y usuarios definidos.
RNF03	Multiplataforma	El sistema deberá funcionar independientemente del sistema operativo
RNF04	Disponibilidad	El sistema estará disponible cuando el usuario lo necesite.
RNF05	Confiabilidad	El sistema presentará información adecuada y congruente
RNF06	Desempeño	El sistema no presentará problemas para su manejo e implementación

2.7 Diseño

2.7.1 Diagramas de casos de uso

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo.

Se presentan los casos de uso para éste Sistema.

“UML is a visual language for modeling and communicating about systems through the use of diagrams and supporting text” [10].

Notación Casos de Uso

En la ilustración 2.1 se muestran los principales elementos que son necesarios para el diseño del sistema utilizando la notación de casos de uso.

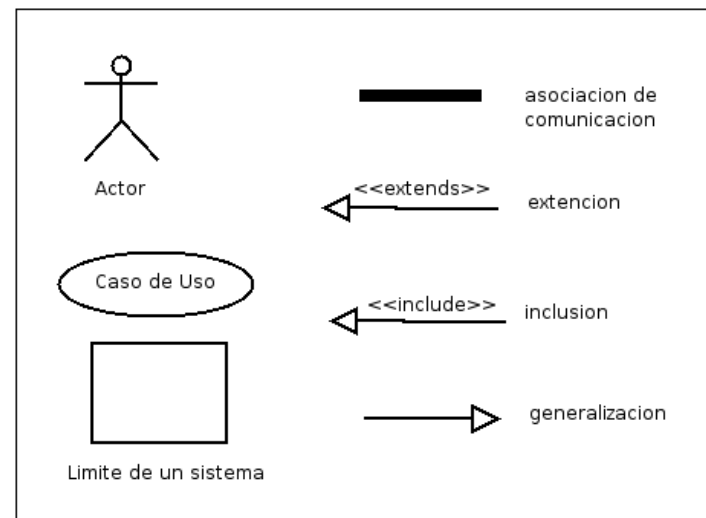


Ilustración 2.1 Notación de casos de uso

En la ilustración 2.2 se muestra de forma general el diagrama de casos de uso del sistema, donde se utiliza la notación descrita en la ilustración 2.2. De esta forma se indica que las principales funcionalidades del sistema en cuestión son:

- Acceso al sistema
- Registro de usuarios
- Rubricas
- Consultas
- Altas de actividades
- Modificaciones de actividades
- Bajas de actividades

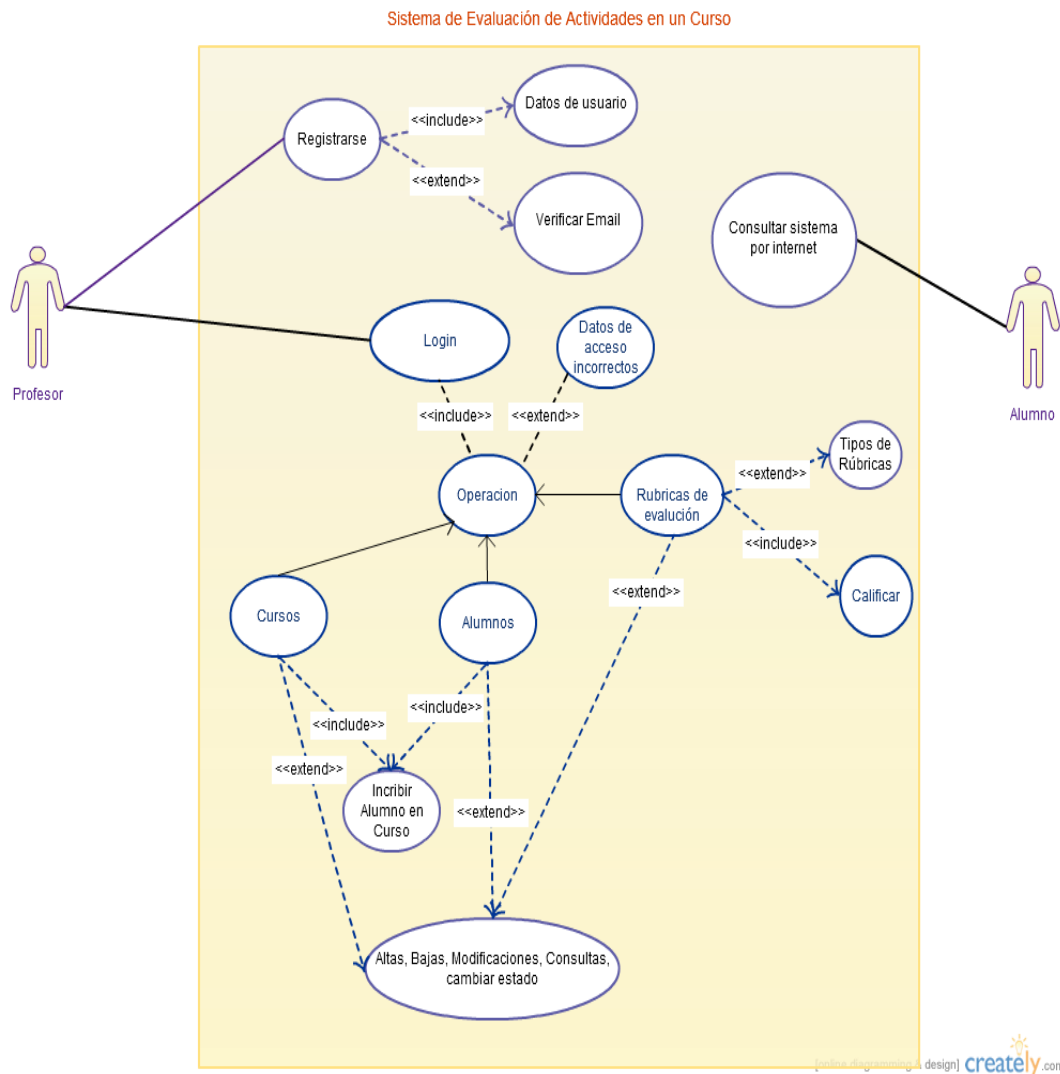


Ilustración 2.2 Casos de Uso del sistema

A continuación se realiza la descripción de cada uno de los casos de uso del sistema, donde se indica el flujo principal para realizar cada una de las acciones.

Caso de uso del Registrarse sistema

2. Descripción del caso de uso

Permite crear una cuenta con un nombre de usuario y contraseña

3. Actor(es)

Profesor		
4. Precondiciones		
ninguna		
5. Postcondiciones		
Se podrá autenticar en el sistema		
6. Pasos (flujo de eventos)		
No.	Acción del actor	Respuesta del sistema
1	Solicitar crear cuenta nueva	Despacha la solicitud y devuelve la hoja de registro nuevo
2	El usuario rellena la información solicitada.	El sistema valida y guarda la información, devuelve la hoja de registro exitoso
3	El usuario ve el mensaje de éxito	
6.1 Pasos Alternativos (flujo de eventos Alternativo)		
1	El usuario intenta registrarse con datos incorrectos o duplicados	El sistema valida la información y devuelve un mensaje de error
2	El usuario visualiza el mensaje de error	
7. Requerimiento asociado		
No hay requerimientos asociados a este caso de uso		
8. Prototipo de interfaz de usuario		
La interfaz a usar es el formulario de registro y algunos mensajes de información		

Caso de uso del Login sistema		
2. Descripción del caso de uso		
Permite autenticar al profesor en el sistema		
3. Actor(es)		
Profesor		
4. Precondiciones		
El usuario debe haberse registrado antes		
5. Postcondiciones		
Se podrá visualizar menú principal		
6. Pasos (flujo de eventos)		
No.	Acción del actor	Respuesta del sistema
1	Ingresa los datos correctos al sistema,	El sistema verifica existencia y correspondencia de usuario y contraseña. Devuelve la hoja de menú principal
2	Ver el menú principal	
6.1 Pasos alternativos (flujo de eventos alternativo)		
1	Ingresa datos incorrectos al sistema	El sistema verifica existencia y

		correspondencia de usuario y contraseña. Devuelve mensaje de error.
2	Visualiza mensaje de error.	
7. Requerimiento asociado		
-Registrarse		
8. Prototipo de interfaz de usuario		
La interfaz a usar es el formulario de login y algunos mensajes de información		

Caso de uso del Gestionar información propia sistema		
2. Descripción del caso de uso		
Permite al usuario modificar sus datos personales		
3. Actor(es)		
Profesor		
4. Precondiciones		
El usuario debe haberse autenticado antes		
5. Postcondiciones		
Visualizar información actualizada		
6. Pasos (flujo de eventos)		
No.	Acción del actor	Respuesta del sistema
1	Rellena el formulario con los nuevos datos	El sistema guarda la información proporcionada y devuelve mensaje de éxito
2	Ver el mensaje de éxito	
6.1 Pasos alternativos (flujo de eventos alternativo)		
1	Cancela modificación de datos	El sistema no guarda ningún tipo de cambio, regresa al menú principal
2	Ve el menú principal	
7. Requerimiento asociado		
-Entrar al sistema		
- Consultar información propia		
8. Prototipo de interfaz de usuario		
La interfaz a usar es la correspondiente a la de datos personales y algunos mensajes de información		

Caso de uso del Altas, Bajas, Modificaciones, Consultas y Cambio de estado sistema		
2. Descripción del caso de uso		
Permite al usuario realizar éstas operaciones con los Cursos, Alumnos, Tipos y rúbricas de evaluación,		

3. Actor(es)		
Profesor		
4. Precondiciones		
El usuario debe haberse autenticado antes, el (los) objeto(s) a administrar debe tener un estado congruente a la operación, por ejemplo, un Curso con estado “terminado” no puede ser modificado.		
5. Postcondiciones		
Visualizar información correspondiente al (los) objeto(s) administrado(s)		
6. Pasos (flujo de eventos)		
No.	Acción del actor	Respuesta del sistema
1	Elige el tipo de objeto a administrar	Consulta: devuelve un listado de los registros encontrados en la BD. Alta: Muestra en el listado el registro añadido. Baja: Desaparece del listado el registro eliminado. Modificaciones: Se resalta visualmente el registro modificado. Cambio de estado: Se muestra el estado actual del objeto.
7. Requerimiento asociado		
Los objetos dependientes deben existir, por ejemplo no puede haber un rúbrica de evaluación sin antes crear una categoría de rúbricas de evaluación (p/e EXAMENES examen1)		
8. Prototipo de interfaz de usuario		
La interfaz a usar es correspondiente a la tabla dinámica (rodeada de menús y operaciones generales) propia del Objeto sobre el cuál se esté trabajando.		

2.7.2 Clases de Usuario y sus Características

Se tienen dos clases de usuarios para este software: Profesor y Alumno.

La persona que desee hacer uso de este software con las funciones básicas para poder visualizar la información del avance que lleva en el curso será a través de la modalidad alumno, con la limitante de no tener acceso de escritura o modificación a la base de datos.

Si se desea acceder al sistema para modificar opciones avanzadas tales como ingresar un nuevo parámetro a evaluar, modificar alguna calificación o modificar de alguna manera la información en la base de datos, tendrá que hacerlo dentro de la modalidad de Profesor.

2.8 Diseño de la Base de Datos

A continuación se presentan los modelos que se han generado del sistema, desde el conceptual, hasta de comportamiento.

2.8.1 Modelo conceptual de la base de datos

A los profesores se les identifica por el número de plaza, y con este deben entrar al sistema. Pueden hacer altas, bajas, modificaciones y consultas a cursos, alumnos, calificaciones y rúbricas de evaluación.

A los alumnos se los identifica principalmente por su matrícula, y se les da continuidad por este atributo.

Las rúbricas se identifican por su clave, se relacionan con cursos y cada rubrica tiene una calificación relacionada con el alumno y el curso en el que se encuentra inscrito.

Los tipos de rúbrica se identifican por su clave, las rúbricas y cada una debe llevar un porcentaje correspondiente a las necesarias para evaluar el curso.

Relaciones

- a) Un PROFESOR imparte un CURSO
- b) Uno o varios ALUMNOS están inscritos en el CURSO
- c) Las RÚBRICAS se clasifican en TIPOS DE RÚBRICAS
- d) El ALUMNO obtiene CALIFICACIÓN en una RÚBRICA de un CURSO

e) A excepción de TIPO RUBRICA y CALIFICACION, todo tiene un ESTATUS

Objetos de datos, atributos y relaciones: el diagrama generado se muestra en la ilustración 2.3.

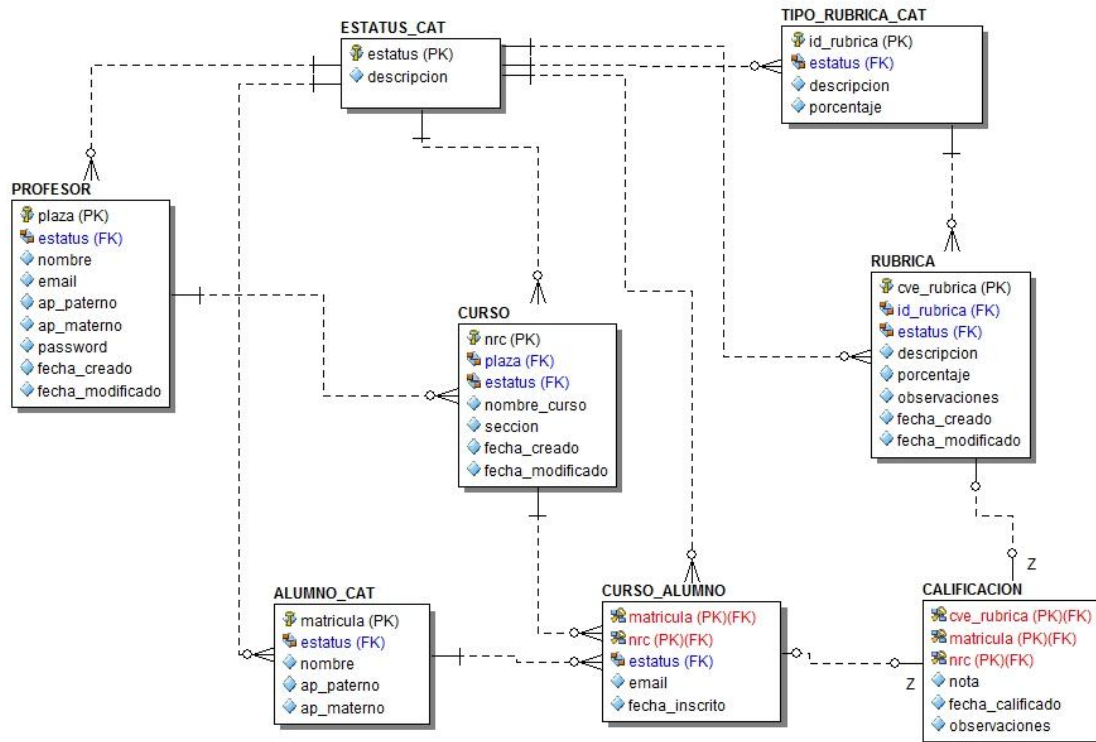


Ilustración 2.3 Objetos de datos, atributos y relaciones

A continuación en la tabla 2.2 definiremos las entidades con sus campos y posibles estados, distinguidos en el Sistema Web que se plantea:

Entidad	Campo	Tipo / Tamaño	Validación	Estados posibles
PROFESOR	Plaza	VARCHAR(20)	Corresponde al identificador único. Clave. Alfanumérico	ACTIVO, INACTIVO
	Nombre	VARCHAR(200)	Corresponde al nombre del usuario que se registra Descripción Campo Fijo Alfanumérico 200	
	Ap_paterno	VARCHAR(80)	Corresponde al apellido paterno del usuario que se registra Descripción Campo Fijo Alfanumérico 80	
	Ap_materno	VARCHAR(80)	Corresponde al apellido materno del usuario que se registra Descripción Campo Fijo Alfanumérico 80	
	Password	VARCHAR(10)	Corresponde a la contraseña del usuario. Clave. Alfanumérico 10	
	EMAIL	VARCHAR(150)	Corresponde al email del usuario Descripción Alfanumérico 150	
	fecha_creado	TIMESTAMP	Corresponde a la fecha del registro del usuario Fecha Campo Fijo Fecha : Formato: DD/MM/AAAA HH:MM:SS	
	Fecha_modificado	TIMESTAMP	Corresponde a la fecha del registro del usuario Fecha Campo Fijo Fecha : Formato: DD/MM/AAAA HH:MM:SS	
CURSO	Nrc	VARCHAR(10)	Corresponde al identificador único.	ABIERTO, CERRADO,

			Clave. Alfanumérico	FINALIZADO
	nombre_curso	VARCHAR(250)	Nombre descriptivo del curso. Descripción. Alfanumérico 250	
	Sección	VARCHAR(5)	Corresponde a la sección de un curso. No requerido	
	fecha_creado	TIMESTAMP	Corresponde a la fecha del registro del usuario Fecha Campo Fijo Fecha : Formato: DD/MM/AAAA HH:MM:SS	
	Fecha_modificado	TIMESTAMP	Corresponde a la fecha del registro del usuario Fecha Campo Fijo Fecha : Formato: DD/MM/AAAA HH:MM:SS	
TIPO_RUB RICA_CAT	id_rubrica	INTEGER	Corresponde al identificador único. Clave. Entero	N/A
	descripcion	VARCHAR(100)	Nombre descriptivo de la rúbrica. Descripción. Alfanumérico 100	
	porcentaje	VARCHAR(10)	Porcentaje de calificación Alfanumérico 10	
RÚBRICA	cve_rubrica	VARCHAR(18)	Corresponde al identificador único. Clave. Alfanumérico	ACTIVO, INACTIVO
	descripcion	VARCHAR(100)	Nombre descriptivo de la rúbrica. Descripción. Alfanumérico 100	
	porcentaje	VARCHAR(10)	Porcentaje de calificación Alfanumérico 10	
	observaciones	Varchar(1000)	Observaciones extra relacionadas con la rúbrica No requerido. Alfanumérico 1000	

	fecha_creado	TIMESTAMP	Corresponde a la fecha de alta Fecha Campo Fijo Fecha : Formato: DD/MM/AAAA HH:MM:SS	
	Fecha_modificado	TIMESTAMP	Corresponde a la última fecha de modificación Fecha Campo Fijo Fecha : Formato: DD/MM/AAAA HH:MM:SS	
ALUMNO	matricula	VARCHAR(20)	Corresponde al identificador único. Clave. Alfanumérico	ACTIVO, INACTIVO
	Nombre	VARCHAR(50)	Corresponde al nombre alumno inscrito en el curso del profesor Descripción Campo Fijo Alfanumérico 50	
	Ap_paterno	VARCHAR(80)	Corresponde al apellido paterno del alumno inscrito en el curso del profesor Descripción Campo fijo Alfanumérico 50	
	Ap_materno	VARCHAR(80)	Corresponde al apellido materno del alumno inscrito en el curso del profesor Descripción Campo fijo Alfanumérico 50	
CALIFICACION	Cve_rubrica	VARCHAR(18)	Corresponde a la rúbrica que se está evaluando Clave Alfanumérico 18	N/A
	Matrícula	VARCHAR(20)	Corresponde al alumno que se está evaluando Clave Alfanumérico 20	
	Nrc	VARCHAR(10)	Corresponde al curso que se está evaluando Clave Alfanumérico 10	
	Nota	VARCHAR(10)	Corresponde a la nota (o calificación) que se otorga al alumno en una rúbrica Alfanumérico 10	
	Fecha_calificado	TIMESTAMP	Corresponde a la fecha de alta	

			Fecha Campo Fijo Fecha : Formato: DD/MM/AAAA HH:MM:SS
	Observaciones	VARCHAR(1000)	Observaciones extra relacionadas con la rúbrica No requerido. Alfanumérico 1000
CURSO_A LUMNO	Matrícula	VARCHAR(20)	Corresponde al alumno que se está evaluando Clave Alfanumérico 20
	Nrc	VARCHAR(10)	Corresponde al curso que se está evaluando Clave Alfanumérico 10
	EMAIL	VARCHAR(150)	Corresponde al email del usuario Descripción Alfanumérico 150
	FECHA_INSCRITO		

Tabla 2.2 Entidades Identificadas, tipos y valores

2.8.2 Diagrama entidad-relación

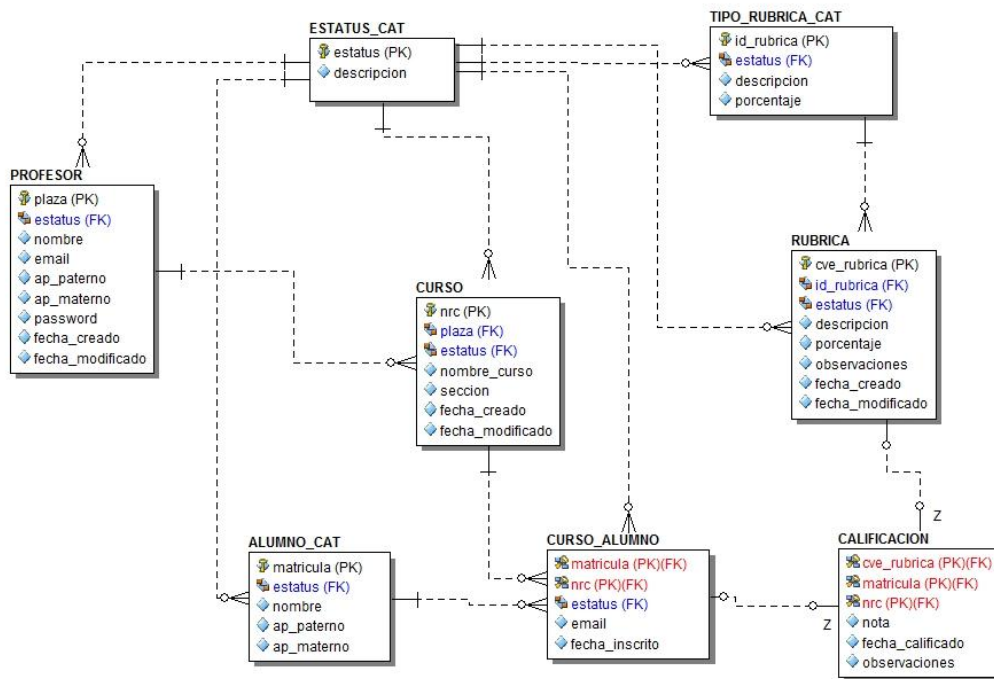


Ilustración 2.4. Diagrama ER

2.8.3 Diagrama de contexto

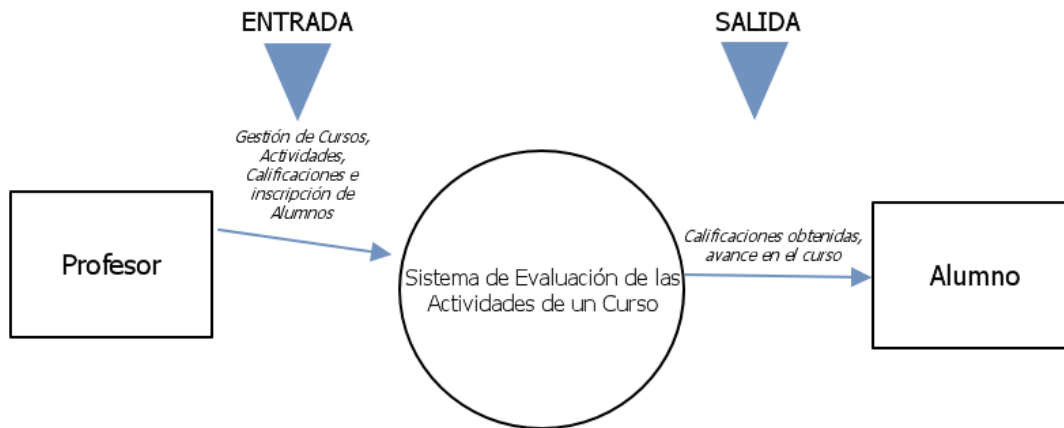


Ilustración 2.5 Diagrama de contexto (Nivel 0)

En el diagrama de la ilustración 2.5 tenemos:

- Entrada: será dada por el Profesor, quien se encargará de administrar los cursos, actividades, calificaciones e inscripción de los alumnos.
- Salida: Los avances, calificaciones y evaluaciones de cualquier Alumno, pueden ser accesadas por los mismos a través de internet.
- Representación del Flujo de información: Como se observa en el diagrama de contexto de la ilustración 2.5, el flujo de información se va alimentar de los datos que introduzca el usuario. Éste debe tener los permisos necesarios para realizar consultas a nuestra base de datos.

Dado que es un proyecto enfocado a Web, se implementa el patrón de diseño Modelo Vista Controlador (MVC) en el sistema (ver ilustración 2.6), por lo que el flujo de la información queda de la siguiente manera:

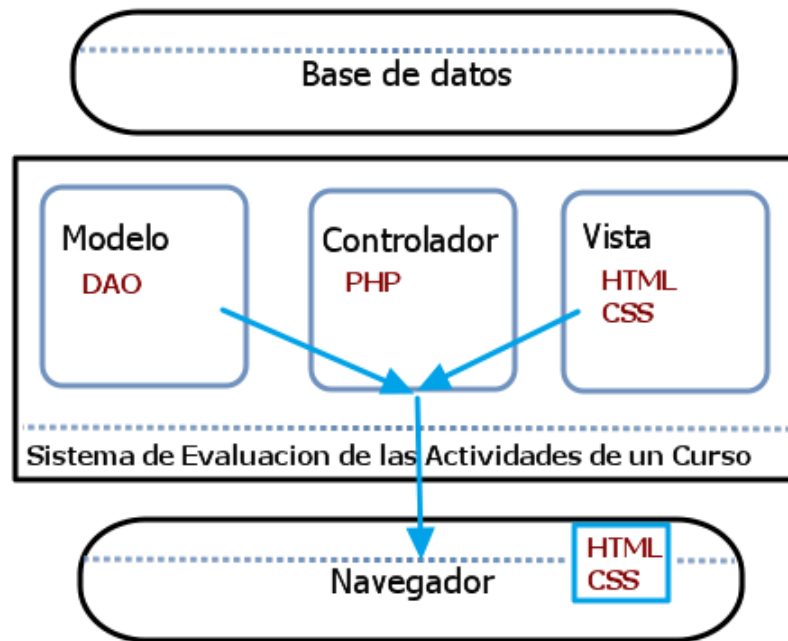


Ilustración 2.6. Flujo de información de patrón MVC

- **Modelo:** Es la abstracción de los objetos a su tabla correspondiente en la base de datos, responsable del acceso a los mismos.
- **Vista:** Es la representación visual de las peticiones al sistema en el lado del cliente (por medio del navegador), HTML y CSS son los usados en éste proyecto Web.
- **Controlador:** Se encarga de la lógica del sistema, en él se definen las reglas de negocio.

2.9 Diagrama de secuencia

Se presenta la secuencia que el sistema sigue para despachar las solicitudes que realiza el usuario. Como puede observarse, el modelo MVC.

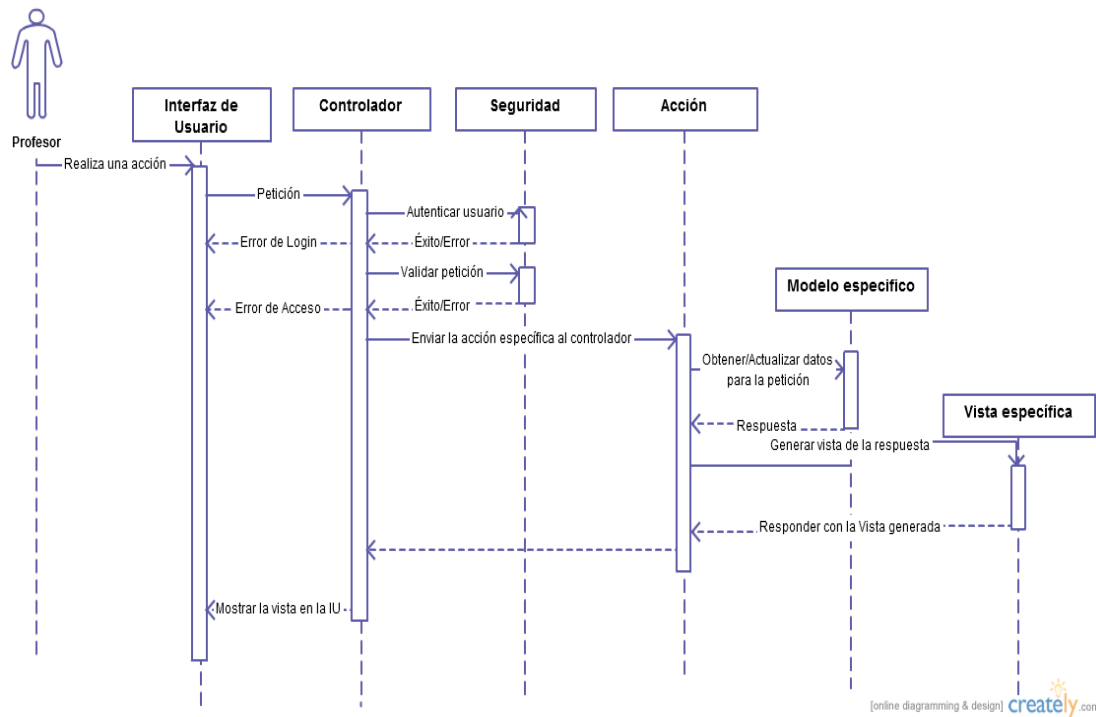


Ilustración 2.7. Diagrama de Secuencia del Sistema

2.10 Diagrama de clases

En la ilustración 2.8 Se muestra el diagrama de clases diseñado para el sistema.

2.11 Métodos de las clases

Clase	Funciones relacionadas
<i>CURSO</i>	-Crear() -Actualizar(id) -Listar() -Borrar(id) -InscribirAlumno(idCurso,idAlumno) -loadModel(id)
<i>ALUMNO_CAT</i>	-Crear() -Actualizar(id) -Listar() -Borrar(id) -loadModel(id) -VerCalificaciones(idAlumno) -EvaluarAlumno()
<i>TIPO_RUBRICA_CAT</i>	-Crear() -Actualizar(id) -Listar() -Borrar(id) -AsignarPorcentaje(idCurso) -loadModel(id)
<i>RUBRICA</i>	-Crear() -Actualizar(id) -Listar() -Borrar(id) -loadModel(id)
<i>CALIFICACION</i>	-Crear() -Actualizar(id) -Listar() -Borrar(id) -loadModel(id)
<i>CURSO_ALUMNO</i>	-Crear() -Actualizar(id) -Listar() -Borrar(id) -InscribirAlumno(idCurso,idAlumno) -loadModel(id)
<i>PROFESOR</i>	-Crear() -Actualizar(id) -Listar() -Borrar(id) -loadModel(id)
<i>ESTATUS_CAT</i>	-Crear() -Actualizar(id) -Listar() -Borrar(id)

-loadModel(id)

2.13 Detalle de métodos de clases

Se describen los métodos (función, entradas y salidas) para las clases definidas.

NOMBRE DEL MÉTODO	Crear()
DESCRIPCIÓN	Crea uno o varios registros en la BD.
ENTRADA	Un arreglo de los modelos a agregar.
SALIDA	Un arreglo con los identificadores de los modelos creados.

NOMBRE DEL MÉTODO	Actualizar ()
DESCRIPCIÓN	Actualiza la información de uno o varios registros en la BD.
ENTRADA	Un arreglo de los modelos a actualizar.
SALIDA	Un arreglo con los identificadores de los modelos actualizados.

NOMBRE DEL MÉTODO	Listar ()
DESCRIPCIÓN	Devuelve un listado de los registros congruente con el tipo de modelo solicitado.
ENTRADA	Ninguno
SALIDA	Un arreglo con los modelos obtenidos de la BD.

NOMBRE DEL MÉTODO	Borrar ()
DESCRIPCIÓN	Elimina uno o varios registros de la BD.
ENTRADA	Un arreglo de los modelos a eliminar.
SALIDA	Un arreglo con los identificadores de los modelos eliminados.

NOMBRE DEL MÉTODO	CambiarEstatus()
DESCRIPCIÓN	Cambia el estatus uno o varios registros de la BD.
ENTRADA	Un arreglo de los modelos a modificar.
SALIDA	Un arreglo con los identificadores de los modelos modificados.

NOMBRE DEL MÉTODO	InscribirAlumno ()
DESCRIPCIÓN	Inscribe a un alumno existente en un curso existente, cambia el estatus del alumno a "INSCRITO"
ENTRADA	El identificador del alumno a inscribir y del curso al

	cual desea inscribir
SALIDA	Un arreglo con los identificadores de los modelos eliminados.

NOMBRE DEL MÉTODO	loadModel ()
DESCRIPCIÓN	Carga un modelo existente de la BD.
ENTRADA	El identificador del modelo.
SALIDA	El modelo obtenido de la BD

NOMBRE DEL MÉTODO	VerCalificaciones ()
DESCRIPCIÓN	Dependiendo el contexto, obtiene las calificaciones respecto a una rúbrica o de un alumno en general.
ENTRADA	El identificador del alumno o de la rúbrica a consultar.
SALIDA	Un arreglo con los identificadores de los modelos consultados.

NOMBRE DEL MÉTODO	EvaluarAlumno ()
DESCRIPCIÓN	Realiza el cálculo de la evaluación total con las calificaciones registradas del alumno
ENTRADA	El identificador del alumno a evaluar
SALIDA	Un arreglo con los identificadores de los modelos consultados.

NOMBRE DEL MÉTODO	AsignarPorcentaje ()
DESCRIPCIÓN	Asigna un porcentaje a una categoría de rúbricas, verifica que el total no exceda el 100%
ENTRADA	Un arreglo con los modelos de categoría de rúbrica a asignar.
SALIDA	Un arreglo con los identificadores de los modelos asignados.

2.14 Interfaces

2.14.1 Organización del menú

Tener en cuenta cada uno de los grupos con las funciones que ejecuta es importante, por lo que a continuación se muestra la estructura final de cada menú principal y sus elementos.

Menú	Vistible	Comentarios
<i>Login/Logout</i>	En cualquier momento	Para iniciar o cerrar sesión.
<i>Mi información</i>	En cualquier momento	Visualizar o actualizar la información del usuario.
<i>Cursos</i>	En cualquier momento	Muestra la interfaz con las operaciones correspondientes, si existen registros los enlista al cargar la página.
<i>Alumnos</i>	En cualquier momento	Muestra la interfaz con las operaciones correspondientes, si existen registros los enlista al cargar la página.
<i>Tipos de actividades</i>	En cualquier momento	Muestra la interfaz con las operaciones correspondientes, si existen registros los enlista al cargar la página.
<i>Actividades</i>	En cualquier momento	Muestra la interfaz con las operaciones correspondientes, si existen registros los enlista al cargar la página.
<i>Calificaciones</i>	En cualquier momento	Muestra la interfaz con las operaciones correspondientes, si existen registros los enlista al cargar la página.
<i>Estadísticas</i>	En cualquier momento	Muestra las opciones para consultar totales por cursos.

2.14.2 Capturas de pantalla del sistema

Se ha tomado un enfoque de tener presentes la mayoría de las opciones en todas las pantallas para una interacción más sencilla con el usuario. Es por ello que los menús y los botones para la gestión de la información son visibles en todo momento desde el acceso al sistema.

2.14.3 Vista general del sistema

Después de cada acción (como eliminar o crear un registro) el sistema devuelve el resultado de la operación en un mensaje que puede ser de tipo éxito, informativo, advertencia y/o error (de datos, no de servidor) (ilustración 11).

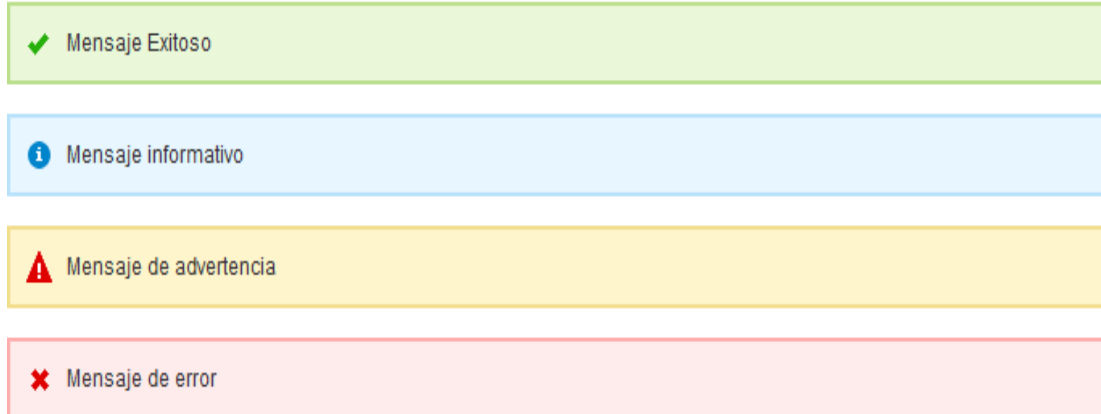


Ilustración 2 Mensajes del sistema

Así mismo, el sistema muestra los errores resultantes de solicitudes no existentes o bien (por parte del, servidor) no permitidas a los niveles de acceso del usuario actual (dado el caso que un profesor quiera modificar la información de otro profesor, o un recurso no encontrado), como se muestra en la siguiente imagen:



Ilustración 3 Ejemplo de Error del servidor

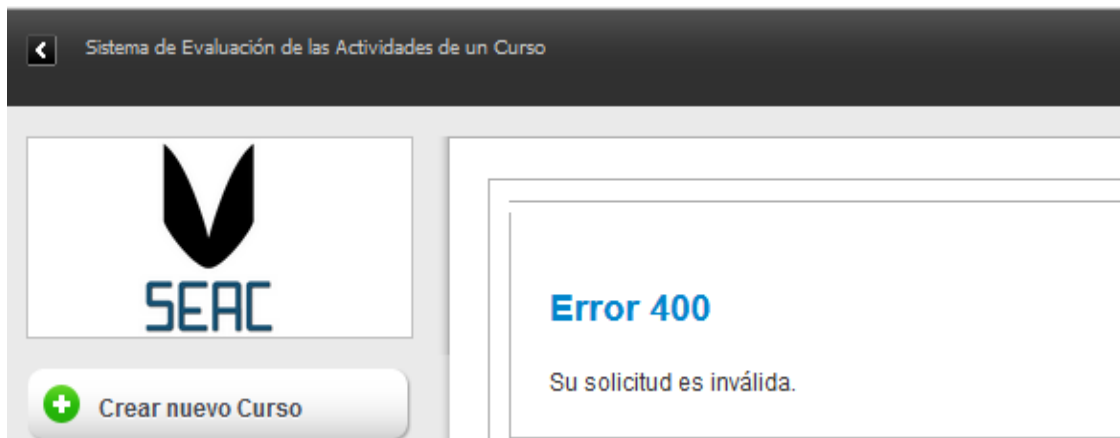


Ilustración 4 Solicitud inválida

En la parte superior y siempre presente el encabezado en el cual el nombre del proyecto visible del lado izquierdo. En la misma sección, del lado derecho el botón de Acceso a usuarios o Salida de sistema (en caso de que exista sesión iniciada).

En la columna izquierda se presenta el logotipo del proyecto con las siglas SEAC, seguido por el área de menú y opciones.

En la parte central se tiene el área para el contenido asociado con la acción o menú a operar, como se muestra en la ilustración 14, el Acceso a usuarios.

Finalmente en la parte inferior se observa el footer, que contiene los derechos de autor y la liga al sitio web del programador del sistema.

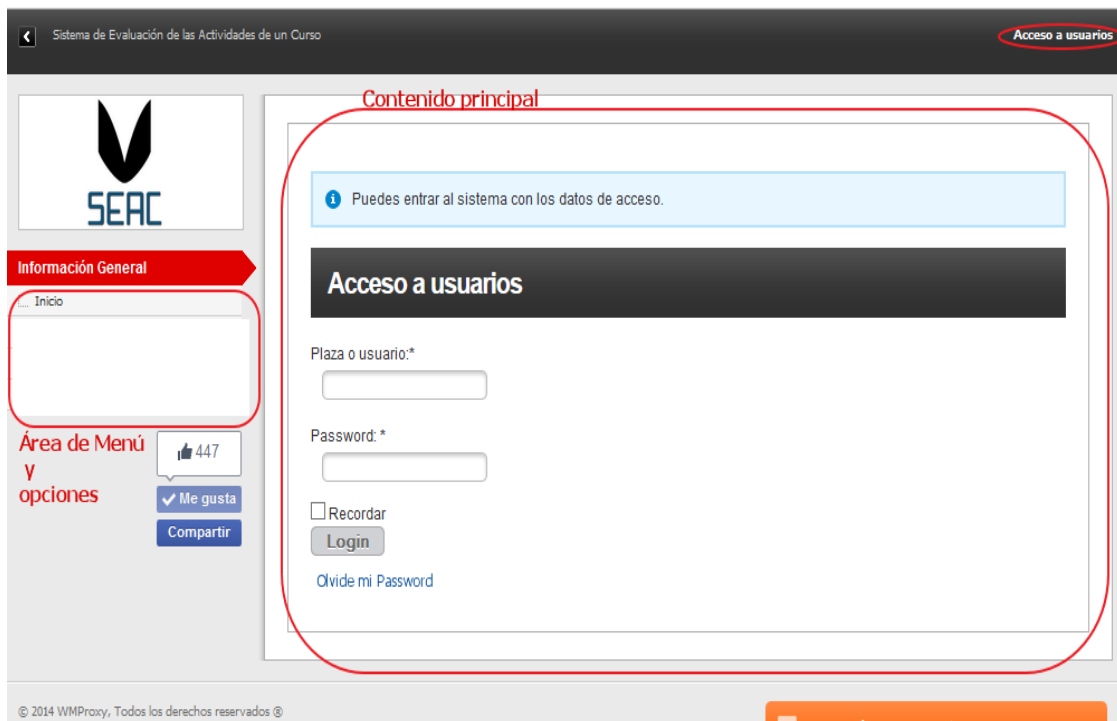


Ilustración 5 Vista General del Sistema (Acceso a usuarios)

2.14.4 Mi información

Se muestra la pantalla de “Mi información”, donde el usuario puede cambiar la información de su registro y perfil.

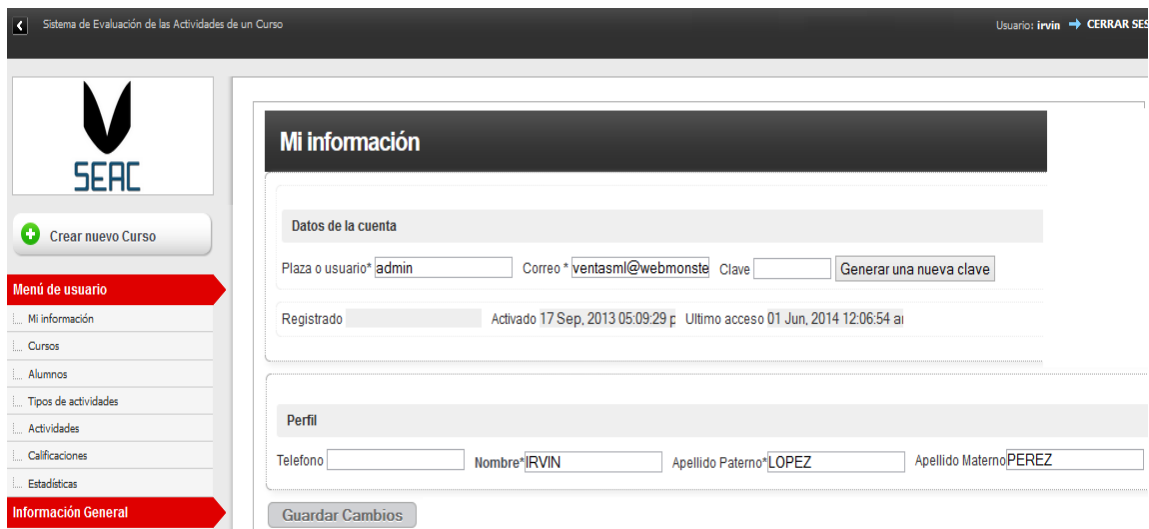


Ilustración 6 Usuario editando Mi información

2.14.5 Gestión de registros

Se muestra un listado de los registros, así como las posibles operaciones correspondientes con cada uno de ellos. La lógica es la misma para los distintos elementos (cursos, alumnos, tipos de actividades y actividades), gracias al framework Kendo UI que facilita las operaciones mediante una tabla dinámica y comunicación asíncrona javascript con el servidor.

Mostrar Ayuda

Aquí puedes CREAR, BORRAR, RENOVAR y PAUSAR Cursos

CURSOS

+ Nuevo

Presionar ENTER para buscar

Curso	NRC	Sección	Fecha Inicio	Fecha Fin				
FISICA II	123456	101	2014-01-24	2014-05-24	Cambiar estado	Editar	Borrar	Actividades

10 por pagina

1 - 1 de 1 resultados

© 2014 WMPProxy, Todos los derechos reservados ©

Ilustración 7 Gestión de Cursos

2.14.6 Estadísticas

El sistema permite acceder a estadísticas de calificaciones de alumnos por curso, mostrado gracias al componente de JQuery especializado para gráficos.



Ilustración 8 Ejemplo de estadísticas para un curso en diversos periodos

Capítulo 3. Construcción e Implementación

3.1 Configuración del Servidor

3.1.1 Infraestructura

Se cuenta con un servidor privado virtual con dirección **IP 198.46.142.225** mediante OpenVZ con las siguientes características:

- 2GB en memoria RAM
- 100GB en disco duro de estado sólido.
- Un límite de 1TB mensual en ancho de banda.
- 2048MB para uso SWAP.
- Procesador Intel E3 4 CPU @ 2.0+ GHz.

3.1.2 El Sistema Operativo: Ubuntu 12.04 LTS

Linux es un sistema operativo robusto, se utiliza en la gran mayoría de servidores web. Al ser de código abierto y gratuito, la documentación y soporte por parte de la comunidad de usuarios ayudan a levantar el servidor sin gran problema, además de reducir los costos invertidos en licencias de uso de software.

“¿Ubuntu se está convirtiendo en la distribución de referencia en el mundo de Linux?”

[11] p. 26

3.1.3 Apache y PHP

La configuración usada en éste proyecto es la que viene de fábrica al instalar Apache y PHP:

- No hay servidores virtuales
- El puerto de escucha web es el 81
- La localización de los archivos del proyecto se encuentra en `/var/www/seac`
- En modo desarrollo PHP se configura para que muestre errores.
- En ambiente productivo PHP no muestra errores.

3.2 Acceso al servidor

Se accede al servidor remoto por medio de SSH, se copian los archivos del proyecto de un ordenador local por medio de clientes SCP.

3.3 Configuración del nombre de dominio y DNS

Mediante un registrador de nombres dominio se obtuvo el nombre webmonster.mx. Se apuntan a los DNS que provee gratuitamente a su vez CloudFlare, un servidor de nombres de dominio distribuido (DNS).

"Cloudflare is able to reduce the load of the web servers and machines by more than 50%"

[12] p. 27

Se ha creado el subdominio **buf.webmonster.mx** para efectos de éste proyecto, el cual apunta a la IP del servidor para el sistema (198.46.142.225).

3.4 Localización de archivos del proyecto

El directorio `/var/www/seac` es el que contendrá los archivos del sistema. Las carpetas principales son:

- Images – contiene todas las imágenes.
- Css – contiene las hojas de estilo.
- Js –contiene todos los archivos JavaScript.
- Protected - contiene directorios y archivos a los que no se tiene acceso via web y son para uso interno del sistema (configuración de la comunicación con la base de datos, modelos, vistas, controladores).

3.5 Permisos de directorios y archivos

Los permisos de los directorios y archivos son los siguientes:

Permisos	Directorio/Archivos
drwxr-xr-x	css
drwxr-xr-x	images
-rw-r--r--	index.php
drwxr-xr-x	js
drwxr-xr-x	protected

3.6 MySQL

- Puerto: 3306
- Motor: InnoDB
- Acceso remoto: desactivado

3.7 Usuario para la Base de datos del Sistema

- Usuario: tesis
- Contraseña: tesis31

- Base de datos: tesis
- Permisos sobre la Base de datos: todos.

3.7.1 PhpMyAdmin

Para un manejo más rápido de la base de datos se utiliza PhpMyAdmin.

- URL: <http://buf.webmonster.mx:81/phpmyadmin>

3.8 Firewall

IpTables es el cortafuegos (o firewall) implementado en el servidor, dado que es el que los sistemas Linux traen por defecto, además, en la industria tiene gran aplicación.

- Regla: todos los puertos están cerrados a excepción del 81 (apache) y 22 (SSH).

3.9 PHP: Yii Framework MVC

Yii es un marco de trabajo que implementa MVC para PHP. Se utiliza en este proyecto gracias a su modularidad, separa cada componente para mejores tiempos de desarrollo y mantenimiento, además de facilitar la abstracción Objeto (nivel programación)-Tabla (capa de base de datos).

“Orientado a objetos, software libre y de alto rendimiento basado en componentes. Algunas de las características de Yii incluyen:

- DAO
- Integración con JQuery
- Entradas de formulario y validación
- Manejo de errores y logging”

-[9] p. 58

Éste framework tiene gran presencia en el ámbito del desarrollo web y tiene una comunidad de desarrolladores bastante grande, gracias a lo cual existe bastante documentación en internet.

Tiene gran cantidad de útiles plugins y extensiones (llamados componentes), se implementan por medio de “widgets”, que son la manera de invocarlos a nivel código.

La utilidad Gii (acceso web) permite crear rápidamente y eficientemente los modelos, vistas y controladores necesarios para operaciones CRUD.

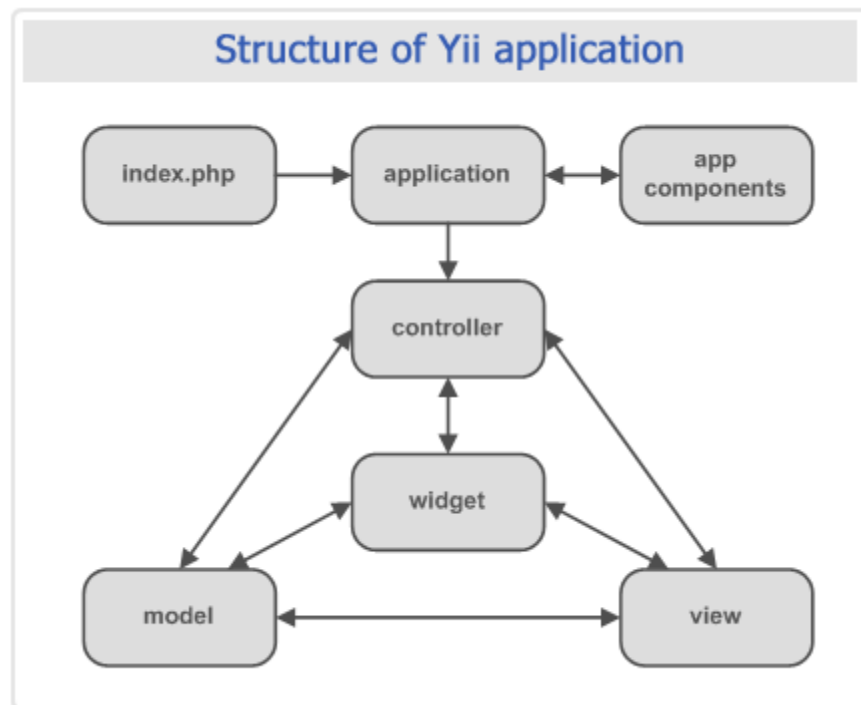


Ilustración 9 Estructura de una aplicación Yii [5] p. 42

3.9.1 Módulo Cruge User

Se implementa el módulo Cruge User para Yii, el cual tiene las siguientes características:

- Manejo de contraseña cifrada a nivel de Base de datos.
- Control de acceso por roles, tareas y operaciones.
- Permite el registro de un usuario vía web, implementando CAPTCHA.
- Confirmación de registro nuevo por correo electrónico.
- Ofrece una sencilla forma de validar el tipo o rol de usuario en el sistema.

3.9.2 JavaScript y HTML5: Framework Kendo UI

Este framework facilita la interacción de la vista con la base de datos. Se utiliza el formato JSON para las respuestas de los resultados de las consultas y gracias a que implementa AJAX no se recarga toda la página web sino que

simplemente se solicitan los datos requeridos. Además, provee automáticamente una interfaz responsiva para aplicaciones modernas, ofreciendo una interacción con gran variedad de dispositivos.

“Web browsers and web standards have matured to the point that today's programmer has rich frameworks to boost productivity.”
[3], p. 18

Capítulo 4. Uso

4.1 Uso del Producto

En esta sección se describen básicamente los requerimientos no relacionados con la funcionalidad del sistema como lo son los volúmenes de usuarios, la disponibilidad, los volúmenes de operación, así como las consideraciones iniciales para la carga de datos y la puesta en marcha de este sistema.

4.2 Disponibilidad

La aplicación estará dando servicio siempre y cuando se encuentre trabajando correctamente la infraestructura de red, de base de datos y del servidor. Esto es, la aplicación podría llegar a estar fuera de servicio en estos horarios estipulados por causas debidas a caídas de funcionamiento de la infraestructura informática de donde se implemente.

4.3 Arranque y Carga Inicial

Para iniciar operaciones con esta herramienta será necesario que el usuario se registre previamente mediante el portal web.

4.4 Características Generales del Producto

Esta sección tiene múltiples propósitos. Presenta un sumario o lista de características generales y adicionalmente los supuestos generales, las limitaciones y riesgos. Finalmente, proporciona una representación detallada de las funciones (o solución a los requerimientos funcionales) desde una perspectiva de usuario final. Todas las reglas de negocio relacionadas con el uso del producto deben estar bien documentadas.

4.5 Funcionamiento Global del Producto

En los siguientes apartados se describen las características generales de funcionamiento del producto como son los supuestos o premisas bajo las cuales se ha elaborado este diseño, los flujos generales de trabajo, así como su representación.

4.6 Supuestos Generales

La infraestructura de comunicaciones que se utilizará será la de la SA para conectarse por medio de http a esta aplicación, independientemente de la plataforma y sistema operativo del usuario.

Al ser ésta una aplicación tipo “web”, habrá diversos comportamientos en la navegación entre páginas/ventanas que son distintas al manejo de una interfaz de

“cliente pesado” o con interfaz clásica de Ms-Windows. Este comportamiento será asimilado fácilmente por todos los usuarios si es que conocen o tienen alguna familiaridad con otras aplicaciones de este tipo.

El diseño gráfico de las páginas web, esto es, los colores, los tipos y tamaños de letras, los menús y todos los demás elementos gráficos son enteramente amigables.

La resolución de la pantalla recomendada para la correcta visualización de datos y navegación a través de las páginas es 1024x768.

4.7 Riesgos y Dependencias

Riesgo de redundancia en bases de datos - Al ser esta una nueva herramienta que operará sobre una nueva plataforma informática es necesario definir una estrategia que evite la redundancia en la nueva plataforma en cuanto a catálogos de cursos y claves de usuarios. En la medida de lo posible se hará que los catálogos estén centralizados y las claves de usuario sean únicas dentro de la nueva plataforma.

4.8 Calidad de datos

Como en toda herramienta informática, la calidad de los resultados, en este caso del registro de tipos de cursos y/o rúbricas de evaluación, depende de que los usuarios alimenten la información de entrada de manera oportuna y correcta.

4.9 Reglas Generales de esta Aplicación

4.9.1 De Captura y Navegación

Cuando se especifica el tipo de dato denominado “alfanumérico” esto significa que letras, números y algunos otros caracteres son permitidos, exceptuando los siguientes:!,[,],{,},^,~,` y “.

Toda la información se ingresa en la base de datos con os in acentos y en mayúsculas y minúsculas.

4.9.2 De Comportamiento

- Los registros solo se graban en la base de datos una vez que han pasado las validaciones correspondientes al pulsarse los botones “Guardar” o “Aceptar”.

- En todas las páginas que conforman el sistema, existen campos que son obligatorios; esto significa que el usuario no puede dejarlos vacíos, sin información. Cuando el usuario presiona el botón “Guardar”, el sistema valida que los campos requeridos contengan datos. De no ser así se desplegará un mensaje con la lista de campos que son requeridos u obligatorios. Las excepciones a esta regla, es decir, cuando los campos no son requeridos, se documentan caso por caso a lo largo de este documento en los apartados denominados “Reglas a Nivel Campo”.
- Todos los campos pasan por una validación cuando el usuario presiona los botones “Agregar”, “Aceptar” o “Modificar”. Básicamente, se valida que el dato introducido por el usuario coincida con el tipo de dato esperado. Las excepciones y validaciones adicionales del proceso de negocio se encontrarán documentadas caso por caso a lo largo de este documento en los apartados denominados “Reglas a Nivel Campo”.

4.9.3 Mensajes de Confirmación / Advertencia y de Error

Los mensajes de advertencia son aquellos que despliega el sistema pidiendo una confirmación al usuario cada vez que este por efectuar una operación que realice una modificación en los datos correspondientes al módulo en el que se encuentra. Los mensajes de error se despliegan cuando ha ocurrido algún evento que impide continuar con el flujo normal.

Conclusiones

A lo largo del desarrollo del presente proyecto se han adquirido y mejorado habilidades inherentes a la ingeniería de software en todas sus etapas. Así mismo, los resultados esperados fueron alcanzados con éxito, ya que como consecuencia se tiene un sistema que cumple con los requerimientos especificados, siguiendo metodologías probadas y adecuadas.

Por otro lado, la adopción de un patrón de diseño como MVC permite que el resolver una problemática sea una tarea más sencilla, coherente y fácil de mantener. El uso de tecnologías nuevas permiten un desarrollo ágil, por lo que es sumamente importante estar al día en las nuevas herramientas de desarrollo que ofrece el mercado (independientemente si son de código privado o abierto).

Mejoras a futuro

Para añadir funcionalidad al sistema, se puede con relativa facilidad realizar una aplicación para dispositivos móviles Android o iOS. Así, no será necesario visitar la página web desde el ordenador, permitiendo compatibilidad con naturalidad en gran cantidad de dispositivos.

También, se podrían agregar módulos de minería de datos, inteligencia de negocios o incluso inteligencia artificial para poder analizar los resultados que obtienen los alumnos para así determinar causas de rendimiento escolar menor y consecuentemente mejorarlo. Aplicando dichas técnicas al sistema, las posibilidades tendrían un gran potencial.

Bibliografía.

- [1]. "MySQL 5: instalación, implementación, administración, programación", Cyril Thibaud, ENI Ediciones, Mayo 2006
- [2]. Learning UML Sinan Si Alhir, O'Reilly, Julio 2003
- [3]. Learning Kendo UI Web Development, John Adams, PACKT Publishing
- [4]. Learning Management System Technologies and Software Solutions for Online Teaching: Tools and Applications, Anthony A. Piña, IGI Global Snippet, 2010
- [5] Yii 1.1 Application Development Cookboo, Alexander Makarov, 2011
- [6]. Ingeniería del Software, Ian Sommerville, Pearson, 2005
- [7]. Fundamentos de jQuery, Rebecca Murphey, Octubre 2011
- [8]. Ingeniería del software: un enfoque práctico, Roger S. Pressman, 1997
- [9]. El nuevo PHP. Conceptos avanzados, Vicente Eslava Muñoz, Bubok Publishing S.L.
- [10]. UML y Patrones, Introducción al análisis y diseño orientado a objetos, Craig Larman, Prentice Hall, 1999.
- [11]. Ubuntu: Administración de un sistema Linux, Gilles CHAMILLARD, ENI Ediciones, Junio 2011
- [12]. Web Performance Daybook Volumen 2, Stoyan Stefanov, O'Reilly
- [13]. Rapid Application Development, MacMillan Publishing Co. James Martin, 1990.