

UNIVERSIDAD DE SONORA

DIVISIÓN DE INGENIERÍA

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Informe Técnico de Práctica Profesional

Partida Ibarra Porfirio Juan Ignacio

Repositorio de Objetos de Aprendizaje

Desarrollo Académico e Innovación Educativa

Hermosillo, Sonora, México a 29 de Junio de 2011.

ÍNDICE

INTRODUCCIÓN.....	4
DESCRIPCIÓN DEL ÁREA DE LA INSTITUCIÓN EN LA QUE DESARROLLÓ LA PRÁCTICA.....	5
JUSTIFICACIÓN	6
OBJETIVOS	7
Objetivo general.....	7
Objetivos específicos	7
PROBLEMAS PLANTEADOS	8
Problemática 1: El Proceso.....	8
Problemática 2: Alternativas Utilizadas	8
Problemática 3: Desarrollo y Evaluación	9
Problemática 4: Redundancia y comunicación	9
Problemática 5: Personal.....	10
ALCANCES Y LIMITACIONES EN LA SOLUCIÓN DE LOS PROBLEMAS	11
Propuestas de Solución de las problemáticas	11
Propuestas de Iniciativas de TI:	11
FUNDAMENTO TEÓRICO	14
PROCEDIMIENTOS EMPLEADOS Y ACTIVIDADES DESARROLLADAS	16
Matriz FODA.....	16
Plan Estratégico de TI.....	17
Misión	17
Visión	17
Objetivo general	17
Objetivos específicos de ti (iniciativas).....	17
Repositorio.....	18
Chat	19
Análisis FODA II.....	20
<i>FORTALEZA</i>	20
<i>DEBILIDAD</i>	21
<i>AMENAZAS</i>	22
Requerimientos técnicos.....	24
Especificaciones de funcionamiento	25

Repositorio de Objetos de Aprendizaje	25
Herramienta de Chat	32
RESULTADOS OBTENIDOS	34
Prototipos	34
Diagramas de secuencias	35
Diagramas de clases	38
Interfaz final	40
CONCLUSIONES Y RECOMENDACIONES	42
RETROALIMENTACIÓN	43
FORTALEZAS Y DEBILIDADES	44
OPORTUNIDADES DETECTADAS	45
REFERENCIAS BIBLIOGRÁFICAS Y VIRTUALES	46
ANEXOS	48
Propuesta	48
Kohana – Breve Guía	49
jQuery – Breve Guía	52
Agregando index.php	54
Apache	54
IIS	54

INTRODUCCIÓN

Con la iniciación del proceso de reconversión de las materias presenciales a materias impartidas en modalidad virtual en la Universidad de Sonora comienzan los trabajos de Unison Virtual (UniVirtual) que mediante su equipo de desarrollo se encarga de la creación de las materias impartidas actualmente en modalidad virtual.

Este proyecto pretende elaborar un espacio permitiendo a los involucrados en la reconversión de materias, equipo de desarrollo, maestros y alumnos el fácil acceso a la ubicación y visualización a los objetos de aprendizaje disponibles así estos conocerán las herramientas a utilizar en las materias a reconvertir ya sea para adaptarse a estas o proponer herramientas nuevas que no se encuentren dentro de las existentes y sean necesarias.

DESCRIPCIÓN DEL ÁREA DE LA INSTITUCIÓN EN LA QUE DESARROLLÓ LA PRÁCTICA

Educación a Distancia (EDUCADIS) es una instancia de Desarrollo Académico e Innovación Educativa de donde surge UniVirtual, área encargada de reconvertir materias de presencial a en línea, así como de impartir los cursos de éstas. En UniVirtual se brinda capacitación a los académicos para formarlos en esta modalidad como instructores. El proceso de reconversión e impartición de cursos en línea consta de siete áreas principales: Autor de Contenido, Diseño Instruccional, Revisión Didáctico Pedagógica, Corrección de Estilo, Diseño Gráfico, Sistemas de Información, Evaluación. El equipo de Sistemas de Información son los encargados de desarrollar o buscar las herramientas y preparar la plataforma donde se imparten los cursos, siendo esta el área en que se realizó la estancia profesional.

JUSTIFICACIÓN

Con la creación de un repositorio de objetos de aprendizaje para las materias de modalidad virtual se facilitará la integración del proceso de reconversión y así eliminar la dependencia a Moodle, lo que permitirá obtener recursos y herramientas personalizadas y enfocadas a las necesidades de UNIVIRTUAL eliminando una gran cantidad estos recursos o herramientas que están almacenados y que no son útiles para el sistema.

Apegar el proceso de desarrollo de software a los estándares existentes del mismo, mediante el uso de tecnologías de información y lenguajes de programación fácilmente adaptables, como son PHP, MySQL, Java y configuraciones basadas en XML para su fácil adaptación a las diferentes plataformas.

Se pretende llevar a cabo una Integración total del proceso de reconversión, por lo cual, es necesario la creación de los recursos y herramientas propias en futuros proyectos.

Entre los recursos a desarrollar, destacan la creación de Chat para las diferentes y temas de discusión, cuya configuración e interacción con la plataforma base se basará en un archivo de XML fácilmente manejable por el equipo de Sistemas de Información.

Con esto se logrará dar pie a la integración de los módulos de la plataforma base, el proceso de reconversión y los recursos y materiales de aprendizaje. Para ello es necesario usar paradigmas de desarrollo escalables, como la programación orientada a objetos.

OBJETIVOS

Objetivo general

El objetivo del proyecto es la creación un repositorio de objetos de aprendizaje para las materias de modalidad virtual que facilite la integración del proceso de reconversión y así eliminar la dependencia a Moodle lo que permitirá obtener recursos y herramientas personalizadas y enfocadas a las necesidades de UNIVIRTUAL eliminando una gran cantidad estos recursos o herramientas que están almacenados y que no son útiles para el sistema.

Objetivos específicos

- Aumentar la productividad del proceso.
- Crear herramientas personalizadas enfocadas a las necesidades de UniVirtual.
- Mejorar la usabilidad del sitio.
- Facilitar la forma en que se agregan las actividades.
- Crear espacios que permitan integrar el proceso de reconversión.

PROBLEMAS PLANTEADOS

Problemática 1: El Proceso

- Creación de documentos digitales

Se realiza la creación del documento, y se distribuye varias veces, creando múltiples copias del mismo.

- Módulos independientes

Al no estar centralizado el funcionamiento de creación del guión instruccional, hace ver que el proceso funciona como si fueran distantes e independientes, cuando realmente están interrelacionados.

Problemática 2: Alternativas Utilizadas

- Adaptación de herramientas y recursos de Moodle.

La instalación de Moodle lleva consigo múltiples herramientas no utilizadas, se instalan módulos que este software ha creado para satisfacer necesidades generales, de manera que llega a ser deficiente en casos específicos de las actividades.

- Integración.

La utilización de un Sistema externo, provoca la dificultad en la integración por parte del equipo de reconversión a la plataforma, así como apegarse a los estándares de desarrollo.

- El equipo de reconversión no tiene acceso a conocer cómo son realmente las herramientas y recursos que solicitan para agregar a los cursos.

- Se requiere personal con experiencia y conocimientos en el manejo de Moodle.

Problemática 3: Desarrollo y Evaluación

- Usabilidad y Accesibilidad.

El desarrollo de la plataforma no ha recibido evaluaciones de usabilidad y accesibilidad, temas que son fundamentales en el diseño y desarrollo, asimismo, no tiene integradas funciones que puedan ayudar al seguimiento y evaluación de la utilización, duración en las tareas, etc.

- No hay proceso de retroalimentación.

El equipo encargado de dar los cursos no proporciona información que sirva al equipo de desarrollo para saber si la plataforma se ha apegado a las características del usuario. Esta problemática crece un poco más con el hecho de que los datos recolectados por Moodle son de difícil procesamiento para evaluar (ya que son datos demasiado distribuidos).

Es bien sabido que un Sistema de Información está hecho para sus usuarios, por lo tanto, es necesario saber qué respuesta ha tenido al utilizarlo, si fue fácil o difícil realizar las tareas, si se apegó a las diferentes necesidades, etc; de manera que el usuario final se concentre en la tarea y no en cómo realizarla.

Cuando se desconoce realmente quién será el usuario, es necesario utilizar estrategias de evaluación de la usabilidad, por ejemplo, someter a alguien que no conoce el sistema y se apega al perfil del usuario final a utilizarlo mientras el equipo de desarrollo observa qué hace para llevar a cabo una lista de tareas asignadas. Otra forma de realizar esto, es que el software recupere datos de los movimientos o acciones hechas para llegar a donde el usuario debe.

Problemática 4: Redundancia y comunicación

- Los materiales didácticos no son compartidos.
- Lecturas, actividades, etc.

Provoca que se repitan muchas veces los recursos y que no se aproveche lo que otros autores de contenido proponen o presentan.

Problemática 5: Personal

Reconvertir una materia de presencial a en línea requiere un equipo relativamente grande, mismo que la Universidad de Sonora no tiene, y se ve obligada a pedir ayuda a externos.

Para crear recursos y herramientas de todas las materias a reconvertir, es necesario un equipo de Desarrollo completamente dedicado a esta actividad, lo cual en este momento no es, sino que se comparten varios roles.

ALCANCES Y LIMITACIONES EN LA SOLUCIÓN DE LOS PROBLEMAS

Propuestas de Solución de las problemáticas.

- Plataforma de trabajo integrado.
- Unificar el proceso de desarrollo.
- Creación del repositorio de objetos (herramientas y recursos) de aprendizaje.
- Basarse en el guión y las funciones utilizadas.
- Pruebas de Usabilidad y Accesibilidad
- Permitir retroalimentación (encuestas, análisis de datos)
- Utilizar el espacio en común.
- Facilitar la obtención de estos materiales.
- Reunir los recursos didácticos.

Propuestas de Iniciativas de TI:

- **Generar proyectos para diferentes departamentos y explotar los equipos y espacio.**
Proyectos semestrales, soluciones a problemas pequeños siguiendo los antecedentes y documentaciones del área de desarrollo principal. Tiempo de respuesta a máximo un semestre dependiendo del proyecto, la necesidad es media, ya que se puede seguir usando lo que hay por algunos años sin aumentar la oferta educativa. El alcance en cambio es amplio, ya que con asesoría de expertos en diversas áreas de estudio se pueden detectar mejor los problemas y oportunidades de mejora en otras áreas.
- **Reclutamiento de personal relacionado con el área de TI.**
El plazo puede ser inmediato, sin embargo, el alcance es meramente el

aumento de la productividad, el costo será elevado al principio, sin embargo a largo plazo se aumentará gradualmente la oferta educativa y si el personal es dedicado a tareas específicas, la eficiencia aumentará incluso en el equipo de desarrollo actual, quienes coordinarían y gestionarían las áreas operativas.

- **Reclutamiento de personal experto en las materias en modalidad virtual.**

El tiempo de respuesta es inmediato, el costo elevado y redituable a largo plazo, es un hecho que se necesita para evitar la dependencia de externos en procesos internos y el alcance será determinado por la especialidad que se opte por reclutar, es necesario investigar qué carreras son exitosas en esta modalidad y tomar pocos riesgos conforme el equipo crece.

- **Realizar proyectos para apoyo en el equipo de desarrollo.**

Las pruebas de usabilidad y accesibilidad son útiles en el desarrollo de software, pese a que no todos los desarrolladores las realizan por considerarlas más costosas que redituables, el hecho es que los cursos en línea no serán eficientes si los errores pasan desapercibidos, por lo cual es necesario hacer pruebas, ya sea en laboratorios o al menos en la capacitación; y de ser necesario utilizar la reingeniería en los procesos que no satisfagan tales pruebas. El alcance es muy amplio si hay “errores silenciosos” en estos momentos. El tiempo de respuesta es largo y el costo es elevado, pero el riesgo de cometer errores en cursos en línea sin estas pruebas es mucho mayor.

- **Integración de los módulos y retroalimentación con los recursos existentes.**

Con tiempo de respuesta de 6 meses con personal capacitado y dedicado en el área de desarrollo de software o 1 año con personal no dedicado únicamente a desarrollo. El alcance es bastante amplio en este punto ya que se está limitado a las herramientas de Moodle y a los conocimientos básicos del equipo de desarrollo en gestión de nuevas tecnologías. Es totalmente necesario que el equipo pueda comunicarse de manera

integral para reducir el tiempo de respuesta en proyectos futuros, ampliar el alcance y reducir la dependencia de otras herramientas.

- **Rediseño con paradigmas existentes de programación.**
Regularmente implícitos en la iniciativa anterior. Facilita en muchos aspectos la integración, la reutilización, establecimiento de normas y formas para trabajar. El Alcance es muy amplio, el tiempo de respuesta es largo por tener que reconvertir código existente.
- **Utilizar una plataforma de trabajo (framework) para unificar al equipo.**
Regularmente implícito en la iniciativa anterior. El Alcance es muy amplio, pero el tiempo de respuesta es largo por tener que reconvertir código existente. Los costos pueden resultar elevados al principio, pero al mejorar gradualmente la escalabilidad de la plataforma y centralizar los módulos, el costo se reduce en proyectos grandes.

FUNDAMENTO TEÓRICO

Para llevar a cabo la selección de la metodología para escoger una iniciativa de TI a desarrollar y obtener de manera puntualizada las problemáticas, fue necesario realizar un Análisis Situacional donde se cubre la situación actual (lo que hace la empresa actualmente) desde diferentes perspectivas para poder determinar el estado en que se encuentra el negocio ^[1], seguido de un Análisis de Fortalezas, Oportunidades, Debilidades y Amenazas (FODA) en el cual se realiza una matriz de comparación que facilita la comprensión de las problemáticas y determinar el impacto de las iniciativas.

UniVirtual es joven como área, y sus objetivos, misión y visión siguen en construcción; sin embargo, uno de los objetivos de DACIE es incrementar la oferta educativa, por lo que los objetivos del proyecto pretenden impactar en éste, ya que un proyecto o área debe tener relevancia con los objetivos del área al que pertenece ^[2]. Finalmente se optó por la iniciativa de TI del Repositorio de Objetos de Aprendizaje.

El proceso para desarrollarlo, se basó en el paradigma de la Ingeniería de Software que cubre las etapas de Análisis, Diseño, Desarrollo, Pruebas, Retroalimentación, Implementación y Mantenimiento vistos en Análisis y Diseño de Sistemas Orientados a Objetos con el Lenguaje Unificado de Modelado (UML)^[3].

Recabando la información del Análisis Situacional y las entrevistas con el equipo de desarrollo, se desglosaron los requerimientos del sistema y de las propuestas de iniciativas se generaron diagramas previos a desarrollo para visualizar el funcionamiento de los cambios al optar por determinada iniciativa, los diagramas fueron básicamente de Clases y Secuencias de UML con sintaxis de java, ya que UML no tiene un consenso en cuanto a sintaxis de código, los diagramas fueron sencillos y concretos para ilustrar lo que se pretendía de manera general^[5]. Prototipos Construidos con el diseño visual destinado, es decir, el contenido web fue convertido a prototipo en HTML, mientras que el prototipo de Chat en Java fue convertido a su versión en aplicación de escritorio ^[4].

El desarrollo del repositorio de objetos como plataforma se hizo utilizando el Framework Kohana de PHP, esto con el fin de seguir el paradigma de la orientación a objetos y poder apegarse al diseño de UML. Por otro lado, la herramienta de Chat se construyó siguiendo una arquitectura Cliente-Servidor a través del protocolo TCP con Multi Hilos.

Por mantenimiento, se regresó a la etapa de Diseño para crear la estructura de la base de datos, que en un inicio no se necesitaba. Así como para agregar funciones para usuarios registrados y colocarlas en una barra de herramientas basada en el diseño por agrupación ^[4].

Se aplicaron los conocimientos de programación abordados en la carrera para los lenguajes de programación PHP y Java, y de las herramientas CSS, HTML y XML como soporte para el repositorio y el chat.

Los manuales de programación fueron elaborados siguiendo un consenso o "buena práctica de programación" donde se coloca la descripción del código justo arriba del mismo con la nomenclatura también conocida como documentación de código fuente ^[6].

Siguiendo con el ciclo de desarrollo de software, se implementó el sistema y se solicitó nombre de dominio, siendo éste: repositorio.univirtual.uson.mx y se brindó asesoría y capacitación al equipo de desarrollo de UniVirtual para comprender el funcionamiento interno del sistema para futuras modificaciones, ya que por utilizar un Framework, el sistema queda con la posibilidad de ser fácilmente escalable.

PROCEDIMIENTOS EMPLEADOS Y ACTIVIDADES DESARROLLADAS

Se colaboró en la realización de un Análisis Situacional que terminó en la propuesta de la solución a las problemáticas anteriormente mencionadas, cuyo resultado fue una matriz FODA (ver tabla 1).

Matriz FODA

	Oportunidades (O)	Amenazas (A)
	<ul style="list-style-type: none"> · Reputación universitaria · Acceso a Informática Administrativa. · Acceso a otros departamentos. · Experiencia del personal externo. 	<ul style="list-style-type: none"> · Políticas universitarias · Integración de personal externo. · Dificultad de integrar módulos Competencia
Fortalezas (F)	Estrategia "Maxi-Maxi"	Estrategia "Maxi-Mini"
<ul style="list-style-type: none"> · Espacios de trabajo. · Equipos de cómputo. · Metodología. 	<ul style="list-style-type: none"> · Generar proyectos para diferentes departamentos y explotar los equipos y espacio. 	<ul style="list-style-type: none"> · Integración de la Metodología y el personal.
Debilidades (D)	Estrategia "Mini-Maxi"	Estrategia "Mini-Mini"
<ul style="list-style-type: none"> · Módulos independientes · Retroalimentación · Poco personal. · Código poco estandarizado 	<ul style="list-style-type: none"> · Reclutamiento de personal relacionado con estatutos universitarios. · Realizar proyectos para apoyo en el equipo de desarrollo. 	<ul style="list-style-type: none"> · Integración de los módulos y retroalimentación con los recursos existentes. · Rediseño con paradigmas existentes de programación. · Utilizar framework para unificar al equipo.

Tabla 1. Matriz FODA.

Se propuso formalmente el proyecto del Repositorio de Objetos de Aprendizaje como plan estratégico de TI.

Plan Estratégico de TI

Misión

Contribuir en la mejora del proceso de reconversión de materias a modalidad virtual mediante un repositorio que permitirá obtener recursos y herramientas personalizadas y enfocadas a las necesidades de UniVirtual eliminando recursos y herramientas que están almacenadas y que no son útiles para el sistema.

Visión

Dentro de los tres años siguientes existirán los recursos suficientes para reconvertir la mayoría de las materias a modalidad virtual.

Objetivo general

Creación de un repositorio de objetos de aprendizaje para las materias de modalidad virtual que facilite la integración del proceso de reconversión y así eliminar la dependencia a Moodle.

Objetivos específicos de ti (iniciativas)

- Aumentar la productividad del proceso.
- Crear herramientas personalizadas enfocadas a las necesidades de UniVirtual.
- Mejorar la usabilidad del sitio.
- Facilitar la forma en que se agregan las actividades

En base a una lista de requerimientos básicos acompañadas de prototipos de baja fidelidad ^[4], se realizaron prototipos visuales de “alta fidelidad” ^[4] donde se hizo uso de HTML, para el repositorio de objetos y la propuesta de plataforma integral; y java para el Chat.

El prototipo del Repositorio (figura 1), permitía navegar entre archivos y mostrarlos. El prototipo de la plataforma integral (figura 2) permitía arrastrar de una caja de herramientas el icono de la herramienta a configurar, por ejemplo, si se arrastraba el icono de MS WORD, se abriría una ventana de configuración

para enlazar el recurso, para ser mostrado como funciona actualmente, esta propuesta en caso de separarse de Moodle, la cual podría ayudar a que no fuera el equipo de Sistemas de Información los que siempre hicieran esto, sino personal con conocimientos básicos. Tras ser aceptado el repositorio, se hicieron cambios para adaptarlo a la vista clásica de un navegador de archivos (figura 3). Siendo el cuerpo del diseño del Chat y de las Propuestas los diagramas de clases (figuras 7 y 9) y de secuencias (figuras 4, 5 y 6).

Repositorio

El repositorio de objetos de aprendizaje fue evolucionando por pruebas de usabilidad y funcionabilidad, ya que esta iniciativa de TI tiene como ventaja competitiva el que es fácil de usar a diferencia de la competencia, así como la pre-visualización de archivos. Lo cual facilitará el proceso de reconversión y a su vez apoyará a aumentar la oferta educativa, objetivo de DACIE.

Con el fin de mejorar la experiencia al usar este sistema, se añadió la función de búsqueda por palabras clave y al compararlo con la competencia, los archivos carecían de metadatos, por lo que se eligió el estándar de metadatos Dublin Core ^[11] como base para agregar al repositorio.

De tal manera que se llevaron a código para el repositorio las siguientes características:

Acceso a la carpeta que contiene los archivos.

Imita el comportamiento de la navegación de archivos acostumbrado.

Se accede al contenido de carpetas y sub carpetas.

Se puede “subir” al directorio anterior.

Se puede especificar la ruta o carpeta deseada.

Se puede visualizar el archivo seleccionado (imagen, lectura, audio, video).

Se recuerda la ruta del último archivo seleccionado.

Permite buscar archivos especificando una palabra clave.

Permite autentificar.

Se muestra una lista de herramientas o actividades existentes.

Se visualiza la herramienta seleccionada en una nueva ventana. Facilita la interacción con el sistema. Funciones de autenticación. Salir del sistema o Cerrar sesión. Subir un archivo. Crear carpetas. Eliminar carpetas. Modificar restricciones de archivos aceptados. Modificar la lista de herramientas. Búsqueda de archivos por etiquetas o palabras clave. "Ficha bibliográfica" (basados en el estándar Dublin Core ^[1]) a los archivos y herramientas. Modificar los metadatos. Mover archivos de carpeta.

Se concluyó el repositorio de objetos con su respectivo manual de programador o técnico, y se modificó la interfaz visual final (figura 12) así como la manera en que se visualizan los archivos (figura 13).

El repositorio fue implementado y es visible desde:
<http://repositorio.univirtual.uson.mx>

Chat

Esta herramienta, se basa en la arquitectura cliente-servidor. Elaborada con el lenguaje de programación Java. De manera general gestiona el envío y recibimiento de cadenas (*Strings*) para realizar acciones, como entrar al chat, salir, crear salas, gestionar, etc.

Ambas clases, Client (figura 10) y Server (figura 8), pese a tener una estructura de datos muy similar, aquella se encarga de desplegar el contenido y ésta de gestionarlo, realizar *broadcasting* o *singlecasting* según sea necesario.

Posteriormente a la implementación, y basado en un Análisis Situacional previo a iniciar con la iniciativa, se realizó un segundo Análisis FODA.

Se llevó a código tanto Cliente como Servidor con las funciones de una herramienta colaborativa, donde se crean discusiones, se participa en ellas, se permiten mensajes privados entre compañeros del grupo y se guardan historial de conversaciones. La interfaz final del cliente (figura 11) se apegó a la interfaz convencional de los programas de mensajería instantánea para aprovechar el modelo mental ^[4] de los usuarios. Mientras que para el servidor se desarrollo una interfaz visual (figura 8) y se dio entrega del código fuente para su futuro mantenimiento.

Análisis FODA II

Para este segundo Análisis FODA, se tomaron en cuenta los módulos donde impacta el proyecto, dejando el resto de oportunidades, amenazas, fortalezas y debilidades de la misma manera en que estaban al inicio del proyecto. En el Análisis FODA 1, se separó en las 6 instancias siguientes:

DACIE

UNISON

Procesos

Recurso

Humano

Entorno (competencia en educación a distancia)

De tal manera que sólo hay impacto en DACIE a través de UniVirtual, específicamente en el proceso de reconversión de materias a modalidad virtual y en el caso del Entorno se analizará en la ventaja competitiva.

FORTALEZA

Metodología de reconversión

Una de las fortalezas previas al proyecto era la metodología de reconversión establecida.

En este caso, el proyecto de Repositorio de Objetos de Aprendizaje ayudó a solidificar aún más esta fortaleza, ya que la metodología, pese a ser un buen plan, no especifica el cómo se realizarán algunos pasos de la reconversión, de

tal manera que los escenarios de comunicación entre los diferentes actores en el proceso realizan varios de sus pasos de manera independiente; con el repositorio en marcha, se podrá establecer una mejor relación entre los autores de contenido, expertos en la materia y el equipo de desarrollo dando a conocer un banco de recursos y experimentar la realidad de las herramientas a través de este sistema. Asimismo, este documento especifica la creación de herramientas propias, siendo este proyecto el que incluye no sólo el repositorio, sino una herramienta de comunicación colaborativa conocida como Chat, diseñada especialmente para usuarios familiarizados con interfaces comunes de este tipo.

DEBILIDAD

Falta de integración en los módulos del proceso.

- Documentos.
- Guiones instruccionales
- Desarrollo de la plataforma
- Usuarios finales

Pese a que la iniciativa de TI que sería una Solución Integral de manera definitiva no es el repositorio de objetos en sí, el proyecto fue elaborado como un sistema del tipo operativo siguiendo una estructura y metodología de desarrollo de software tal que se apegue al prototipo de la Solución Integral para que cuando sea factible, no sea necesario rediseñar este módulo ni desecharlo.

El repositorio reducirá la falta de integración en el proceso de la creación del guión instruccional permitiendo compartir documentos entre personas que no son del equipo de desarrollo directamente y eliminará la redundancia de archivos en el desarrollo de la plataforma.

Asimismo, esta iniciativa de TI permite la interacción en una misma plataforma entre los expertos en la materia y el equipo de desarrollo, lo cual hace surgir

una nueva gama de oportunidades de mejora y reducir la amenaza del entorno por no contar con un sistema integral en este proyecto.

AMENAZAS

- Retroalimentación
Después del proceso de reconversión de la carrera de Trabajo Social que se está impartiendo actualmente no existe ningún tipo de retroalimentación hacia el equipo de desarrollo de las materias en modalidad virtual.
- Plataforma UNIVIRTUAL
 - HTML y CSS sin estandarizar.
 - PHP estructurado.
 - Sin pruebas de usabilidad y accesibilidad.
 - Sin retroalimentación
 - Sin documentación del software (manuales de programador, diagramas de clases, de casos de uso, secuencias, etc.).
 - Sin ninguna herramienta propia (desventaja con respecto al entorno).

En este caso, se da inicio a la creación de herramientas propias de la Universidad de Sonora después de 2 años de iniciado el proyecto. Se comienza la creación de manuales y documentación de software, se marca la pauta para dejar de lado la programación estructurada, y al igual que en la ventaja anteriormente mencionada, se generan más oportunidades de mejora, dar apertura a nuevos proyectos con tiempo de respuesta corto y alto alcance.

Implementada esta iniciativa y la propuesta de Solución Integral (ver anexo) la matriz FODA conserva algunos de los puntos anteriores.

	Oportunidades (O)	Amenazas (A)
	<ul style="list-style-type: none"> · Reputación universitaria · <u>Acceso a otros departamentos.</u> Experiencia del personal externo. 	<ul style="list-style-type: none"> · Políticas universitarias · Integración de personal externo. Dificultad de integrar módulos Competencia

<p>Fortalezas (F)</p> <ul style="list-style-type: none"> · Espacios de trabajo. · Equipos de cómputo. · Metodología. 	<p>Estrategia "Maxi-Maxi"</p> <ul style="list-style-type: none"> · <u>Generar proyectos para diferentes departamentos y explotar los equipos y espacio.</u> 	<p>Estrategia "Maxi-Mini"</p> <p>Integración de la Metodología y el personal.</p>
<p>Debilidades (D)</p> <p>Módulos independientes</p> <ul style="list-style-type: none"> · Retroalimentación · Poco personal. · Código poco estandarizado 	<p>Estrategia "Mini-Maxi"</p> <ul style="list-style-type: none"> · Reclutamiento de personal del área de desarrollo de software. · <u>Realizar proyectos para apoyo en el equipo de desarrollo.</u> 	<p>Estrategia "Mini-Mini"</p> <ul style="list-style-type: none"> · Integración de los módulos y retroalimentación con los recursos existentes. · <u>Rediseño de la plataforma principal, iniciativa de TI propuesta en anexo.</u>

Tabla 2. Matriz FODA II.

En las iniciativas de TI propuestas para el segundo Análisis FODA destacan 3 de las propuestas de iniciativas del Análisis FODA 1; lo cual indica que hemos dado un paso adelante pero aún queda mucho por hacer.

La reconversión de la plataforma es sin duda la mejor opción de las iniciativas e incluso el mismo equipo de desarrollo notó beneficios dentro de la propuesta y prototipos como la repartición de tareas a quien realmente se le debe asignar, ya que actualmente es el equipo de desarrollo quien tiene que realizar funciones que son sencillas, pero difíciles para personal ajeno a TI's. Por otro lado la generación de proyectos aprovechando la estancia en la Universidad de Sonora es sin duda una gran ventaja que debe explotarse en pro de crecimiento del proyecto principal, que es la educación a distancia y de situar a la Universidad de Sonora en un estado de ventaja competitiva.

Se elaboró un manual técnico para el equipo de desarrollo que incluye los siguientes apartados:

Requerimientos técnicos

Este sistema está basado en Web, por lo que es necesario que esté montado en un servidor con acceso remoto.

Requiere tener instalada la versión 5.5 de PHP cuyo requerimiento es estar instalado en un ordenador con el servicio de Apache o de IIS (CGI).

Como la interacción se da entre el Cliente (con JQuery) y el Servidor (con PHP), es el cliente quien debe tener un navegador que soporte funciones de Javascript.

El Repositorio de Objetos de Aprendizaje pretende contener una gran cantidad de archivos, por lo que se recomienda instalar en un disco duro lo suficientemente grande para contener los archivos deseados a mostrar; puede ser desde una carpeta con archivos de texto plano hasta cientos de carpetas con imágenes y videos, el sistema en sí no se ve limitado por problemas de espacio; por sí mismo, el repositorio de objetos de aprendizaje tiene un espacio en disco de 4 MB.

Al utilizar Kohana Framework, se requiere que la carpeta `"/repositorio/application/cache/"` y `"/repositorio/application/logs/"` tengan el modo de acceso sin restricciones a lectura y escritura, ya que ahí guarda algunos datos temporales. Por este motivo se recomienda tener acceso y privilegios directos o por escritorio remoto.

El sistema además, requiere privilegios de lectura en la carpeta `/repositorio/media/files` para guardar y leer los archivos dentro.

Por motivos de usabilidad, el sistema puede integrar un archivo que renombre la dirección solicitada. En el caso de apache se utiliza el archivo `.htaccess` y para IIS es el `web.config` (ver Anexo); en cualquier caso, sólo se solicita agregar `index.php` a cualquier solicitud. Ya que como Kohana recibe sus solicitudes a través de `index.php` (ver Anexo) en lugar de escribir o acceder `"/repositorio/index.php/qwerty/"` se usaría directamente `"/repositorio/qwerty/"`

La herramienta de chat, requiere que el sistema permita la ejecución del lenguaje Java así como tener habilitado el puerto necesario para el mismo (se puede cambiar), es una aplicación soportada por sí misma (stand-alone) capaz de funcionar a través de consola o interfaz gráfica. El servidor por sí mismo requiere 140 KB en disco y al menos 40 MB de Memoria RAM para ejecutar la máquina virtual de Java más el servicio de Chat, el cliente requiere lo mismo que el Servidor para su ejecución, y sólo necesita comunicarse al puerto en que el Servidor esté funcionando. Se recomienda tener instalada la versión más actualizada de la máquina virtual de Java (6 hasta el momento).

Especificaciones de funcionamiento

Repositorio de Objetos de Aprendizaje

El Repositorio de Objetos de Aprendizaje utiliza como lenguaje de programación principal PHP, lenguaje que se ejecuta del lado del servidor. Se apoya del lenguaje de script del lado del cliente Javascript, para su visualización HTML y para su estilo CSS.

Para llevar a cabo las funciones que amplían la usabilidad del sitio, se optó por usar el Framework de Javascript JQuery. jQuery es una biblioteca o framework de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

Se jugarán dos roles diferentes en este sistema, uno es usuario y el otro es de usuario registrado (o con privilegios). La única diferencia entre usuario registrado y no registrado, es que el primero tiene la función de subir archivos.

El sistema adopta maneras de trabajar propias para ordenar cosas como:

Qué archivos mostrar.
Qué usuarios acceden.
Qué herramientas se muestran.
Dónde se encuentran los archivos de:
javascript
Css

Imágenes

Iconos de archivos y carpetas.

En general, entendiendo este sencillo funcionamiento, el resto corresponderá a comprender cómo se trabaja con el Framework Kohana (ver Anexo) y el Framework de Javascript JQuery (ver Anexo).

Kohana utiliza como interfaz entre el cliente y el sistema al archivo `index.php` del directorio raíz. Con el fin de mantener el orden, evitar redundancia de archivos y facilitar el respaldar o mover el sistema, se emplea el uso de rutas absolutas dinámicas, quizás suene extraño usar rutas absolutas y facilitar mover sistemas completos, el hecho es que cuando se trabaja con archivos, no se permite utilizar rutas relativas, pero PHP nos permite definir variables globales que se comportan como *variables de entorno*, de tal manera que si ese archivo (`index.php`) se comunica siempre con todo el sistema dentro del Framework, entonces podemos definir en él las variables absolutas que necesitamos. De manera predeterminada, Kohana nos define al menos 4 variables de entorno, las cuales apuntan hacia directorios sobre los que corren las aplicaciones, módulos y el sistema (kohana, no el repositorio). El sistema requerirá saber dónde tenemos nuestros archivos de Javascript, nuestros archivos del repositorio, imágenes e iconos. Lo que se necesita es buscar el código siguiente (o similar):

Dirección real del repositorio,

```
DEFINE('URL', "http://www.univirtual.uson.mx/ ");
```

Valor de la ruta absoluta al sitio para IIS

```
DEFINE('ABSOLUTE', "C:/inetpub/wwwroot");
```

para Apache (wamp)

```
DEFINE('ABSOLUTE', "C:/wamp/www");
```

El directorio raíz de Kohana

```
DEFINE('ROOT','/repositorio');
```

El directorio donde incluiremos archivos de JS, CSS, Imágenes, XML, etc.

```
DEFINE('MEDIA',ROOT.'/media');
```

La ubicación de archivos de JavaScript

```
DEFINE('JAVASCRIPT',MEDIA.'/javascript');
```

Ubicación de archivos de estilo CSS

```
DEFINE('CSS',MEDIA.'/css');
```

Directorio con las imágenes del sistema (background, etc.)

```
DEFINE('IMAGES',MEDIA.'/images');
```

Directorio con los ICONOS para los archivos.

```
DEFINE('ICONS',IMAGES.'/icons');
```

Archivos de configuración XML

```
DEFINE('XML',MEDIA.'/xml');
```

Reproductor multimedia de Flash

```
DEFINE('FLASH',MEDIA.'/flash');
```

Carpeta de archivos del repositorio de objetos.

```
DEFINE('REPOSITORY',ROOT.'/media/files');
```

```
DEFINE('UPLOAD',DOCROOT.'/media/files');
```

```
DEFINE('DIR_REPOSITORY',ABSOLUTE.REPOSITORY);
```

No es impedimento guardar las cosas en diferentes directorios, pero de esta manera se permitirá al equipo de desarrollo tener organizados los archivos en base a su función. De esa manera, los archivos se podrán referenciar en todos los documentos necesarios.

Por ejemplo, en JavaScript utilizaríamos:

```
<script src="<?php echo JAVASCRIPT; ?>/script1.js"></script>  
<script src="<?php echo JAVASCRIPT; ?>/script2.js"></script>
```

Para mostrar imágenes:

```
  

```

Los iconos en los archivos son tomados de la carpeta definida en la variable de entorno ICONS, de tal manera que se define en esa carpeta el nombre de la extensión en archivos .PNG (de preferencia en tamaño de icono 22x22 pixeles aproximadamente), en caso de no existir, se llamará a "unknown.png" de la misma carpeta.

Por ejemplo: "images/icons/jpg.png" será el icono para los archivos del tipo ".jpg"

"images/icons/unknown.png" será el icono para archivos cuya extensión no esté definida en el directorio de iconos.

Para los iconos en las carpetas se deberán agregar dentro de la carpeta a representar archivos con el nombre "_icon.png".

Por ejemplo "Materiales/_icon.png" será el icono para la carpeta "Materiales".

También se utilizarán valores por defecto, en caso de que no exista el icono para el folder o el tipo de archivo, por ejemplo: "images/icons/default.png" será el icono para carpetas que no contengan "_icon.png" en sus archivos.

En el caso de las herramientas, se llenarán *alimentadas* del archivo XML ubicado en la dirección XML de variable de entorno. El archivo tiene por nombre "tools.xml" y su contenido será definir el total de herramientas a mostrar en el repositorio de objetos con su nombre, descripción, lenguaje o tipo de herramienta, ruta o ubicación del archivo de prueba y el icono a desplegar. La razón por la que se pide la ruta e icono de la herramienta, es porque esta sección será independiente del repositorio, aunque puede formar parte del

framework, puede resultar que ciertas herramientas sean externas o su funcionamiento se vea en conflicto con el del sistema mismo.

XML es un meta lenguaje, por lo que esta manera es de las más efectivas y eficientes para relacionar sistemas independientes por naturaleza del lenguaje utilizado o por restricciones en su funcionamiento. La integración de aplicaciones y las soluciones integrales son sinónimo de utilizar XML o algún framework de este metalenguaje. El sistema buscará dentro de este archivo por todas las etiquetas que coincidan con <tool> para explotar sus tags internos.

De tal forma que formato es el siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<tools>
  <tool>
    <src>URL de la herramienta</src>
    <icon>URL del icono de esta herramienta</icon>
    <id>Id</id>
    <title>Título</title>
    <creator>Autor</creator>
    <publisher>Editor</publisher>
    <date>Fecha de creación</date>
    <type> Tipo de herramienta </type>
    <description>Descripción</description>
    <contributor> Contributor </contributor>
    <format>Format </format>
    <subject>Tema y palabras clave (tags)</subject>
    <source> Fuente de la que se deriva esta herramienta </source>
    <language>Idioma</language>
    <relation>Relacionado con.</relation>
    <coverage>Covertura</coverage>
    <rights>Derechos</rights>
  </tool>
</tools>
```

Al igual que los archivos del repositorio, se basa en el estándar Dublin Core ^[11].

Nota: Las herramientas se abrirán en ventanas nuevas haciendo referencia al vínculo especificado en SRC. Si se encuentran fuera del sitio deberá especificarse la ruta completa (<http://www.misitio.com>).

El sistema realiza una búsqueda de archivos y carpetas de manera recursiva, los usuarios registrados son quienes pueden realizar la función de subir archivo, sin embargo, las carpetas pueden ser restringidas a cierto tipo de archivos a través de un archivo de configuración XML llamado "upload.xml" situado dentro de cada carpeta que se quiera limitar, por ejemplo, para restringir que en la carpeta "/Materiales/Documentos/Articulos" se suban archivos que no sean "PDF", "DOC" o "DOCX" el archivo "upload.xml" deberá especificar tales formatos, un mensaje de error y estar situado en el directorio anteriormente mencionado; si se desea evitar que se suban archivos completamente y limitar el directorio al contenido que ya tiene (útil para forzar a usar una subcategoría), como la carpeta "Materiales" tenemos que colocar un archivo "upload.xml" en tal directorio pero sin especificar ningún formato.

Para permitir sólo pdf, doc y docx:

```
<?xml version="1.0" encoding="utf-8"?>
<allowed_files>
  <file>
    .pdf
  </file>
  <file>
    .docx
  </file>
  <file>
    .doc
  </file>

  <restrict_message>
    Lo sentimos, en este espacio sólo se permite subir archivos con
    extensión ".PDF "
  </restrict_message>
</allowed_files>
```

Para NO permitir ningún archivo:

```
<?xml version="1.0" encoding="utf-8"?>
<allowed_files>
  <restrict_message>
    Lo sentimos, no se permite subir nada en esta carpeta.
  </restrict_message>
</allowed_files>
```

Ya que al existir el archivo "upload.xml" se restringe el subir archivos, se busca entre las etiquetas "<file>" para limitar los formatos permitidos

<allowed_files> Conjunto de formatos permitidos.

<file> Formato permitido

<restrict_message> Mensaje en caso de fallar por restricción.

NOTA: restrict_message es un tag obligatorio, file es opcional.

Herramienta de Chat

Esta herramienta es un doble software, cliente y servidor, ambos desarrollados completamente en Java. Su funcionamiento es básicamente una comunicación clásica de esta arquitectura.

El servidor se pone en estado de escucha en determinado puerto.

Espera por clientes.

Si alguien se conecta, crea una instancia de cliente y abre canales de comunicación.

Sigue escuchando por nuevos clientes hasta finalizar el servicio.

El cliente en cambio, inicia una conexión (Socket) en el puerto y host especificados (en donde debe estar el servidor escuchando). De tener éxito inicia el proceso.

El cliente se conecta.

Establece canales de comunicación.

Envía las cadenas de texto que el usuario ingrese al servidor con formato especial.

Recibe las cadenas por parte del servidor y las interpreta.

Finaliza la conexión.

Los métodos dedicados a la parte de interpretación son "handle" en ambas clases.

Lista de cadenas reservadas para funciones:

```
private static final String helpCommand = "-help";
private static final String categoryCommand = "-category";
private static final String discussionCommand = "-discussion";
private static final String dropCategoryCommand =
"-drop_category";
private static final String dropDiscussionCommand =
```



```

"-drop_discussion";
    private static final String addUserToCategoryCommand =
"-add_user_category";
    private static final String addUserToDiscussionCommand =
"-add_user_discussion";
    private static final String privateCommand = "-pm";
    private static final String broadcastCommand = "-bc";
    public static final String allowCommand = "-allow";
    private static final String renameCommand = "-rename";
    private static final String guestCommand = "-guest";
    private static final String forceLoginCommand = "-forcelogin";
    private static final String usersCommand = "-users";
    private static final String kickCommand = "-kick";
    private static final String banCommand = "-ban";
    private static final String unbanCommand = "-unban";
    private static final String exitCommand = "-exit";

    private static final String hiString="/ping";
    private static final String hoString="/pong";
    private static final String renameString="/rename ";
    private static final String renameString_="[RENAME] ";
    private static final String finalRenameString="/final_rename ";
    private static final String pmString="/pm ";
    public static final String sendPmString = "[PM] ";
    public static final String yourPmString = "[OWNPM] ";
    private static final String exitString="/exit";
    private static final String helpString="/help";

    public static final String guestString="GUEST_";
    public static final String askUsers="[USERS]";
    public static final String askDiscussions="[DISCUSSIONS]";
    public static final String discussionCastString="[DISCUSSION]";
    public static final String userString="[USER]";
    public static final String serverString = "[SERVER] ";
    public static final String removeUserString = "[REMOVE] ";
    public static final String offlineString = "[OFFLINE] ";

```

RESULTADOS OBTENIDOS

Prototipos



Fig. 1 Prototipo en HTML del Repositorio de Objetos



Fig. 2 Prototipo HTML de la plataforma Integral



Fig. 3 Segundo Prototipo HTML del Repositorio de Objetos

Diagramas de secuencias

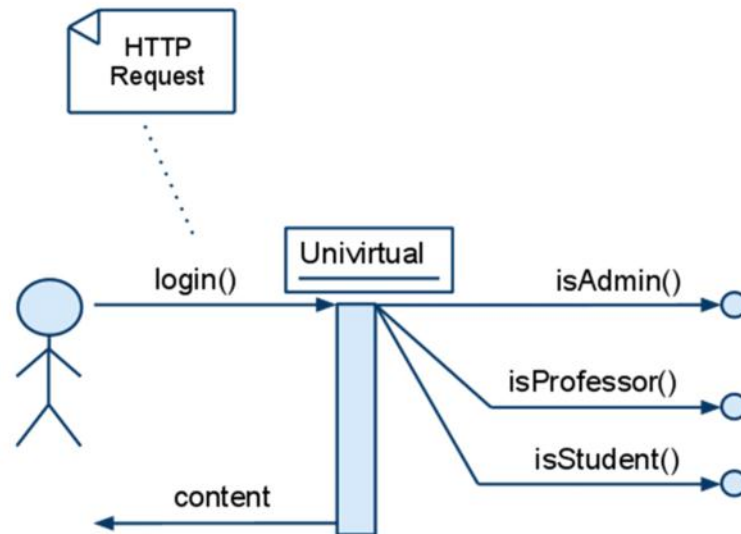


Fig. 4 Diagrama de secuencias login.

El funcionamiento de ingresar al sistema sería el mismo en pasos para el usuario (figura 4).

El usuario con privilegios de agregar recursos (figura 5) y actividades dará clic sobre el objeto, lo arrastrará a donde quiere que aparezca y editará a su conveniencia el nombre y configuración (si se necesita). Apoyado mediante un archivo XML para tomar acciones. Como sabemos, XML es un meta lenguaje y sirve como interfaz entre diferentes lenguajes y plataformas.

Login View: *HTML* + *HTTP Request*

Si se convirtiese la plataforma, el acceso al sitio sería idéntico. El usuario no sentiría ningún cambio, excepto quizás por la url. En vez de utilizar un clásico:

www.misitio.com/login.php

probablemente

www.misitio.com/login

utilizaría:

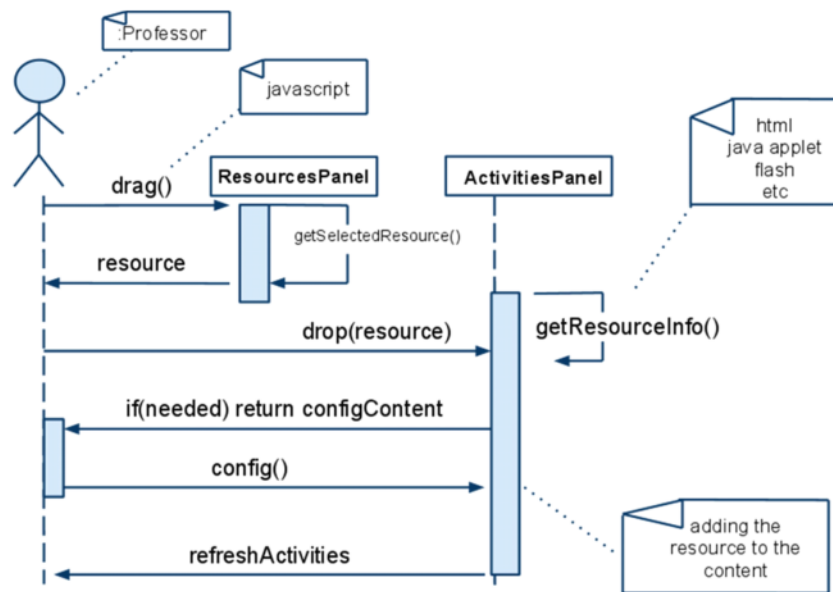


Fig. 5 Diagrama de Secuencia addResource

Activities View: Javascript only
 De manera general cómo se agrega cualquier actividad o recurso (figura 5).

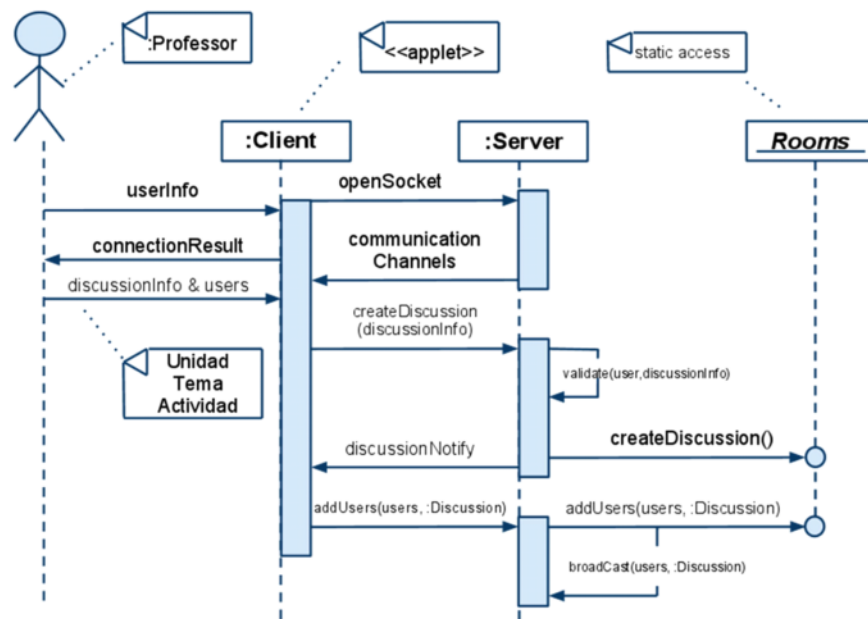


Fig. 6 Diagrama de Secuencias chatConfig

Chat View: *PHP* + *Java* *Applet*

El usuario es un argumento mandado al applet (<param>) mediante una variable de sesión \$_SESSION["user"] o como el framework lo utiliza \$this->get("user");

Específicamente cómo se agrega el chat después de ser llamado (figura 6)

Diagramas de clases

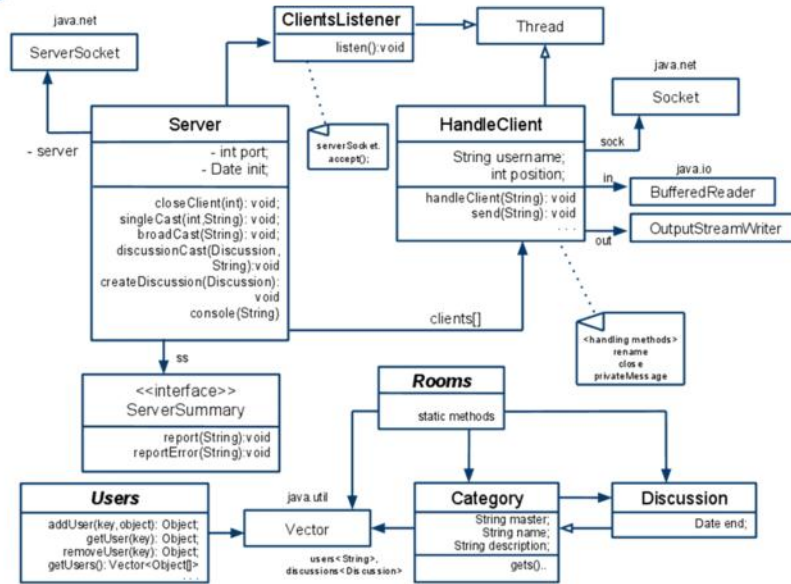


Fig. 7 Diagrama de clases Server

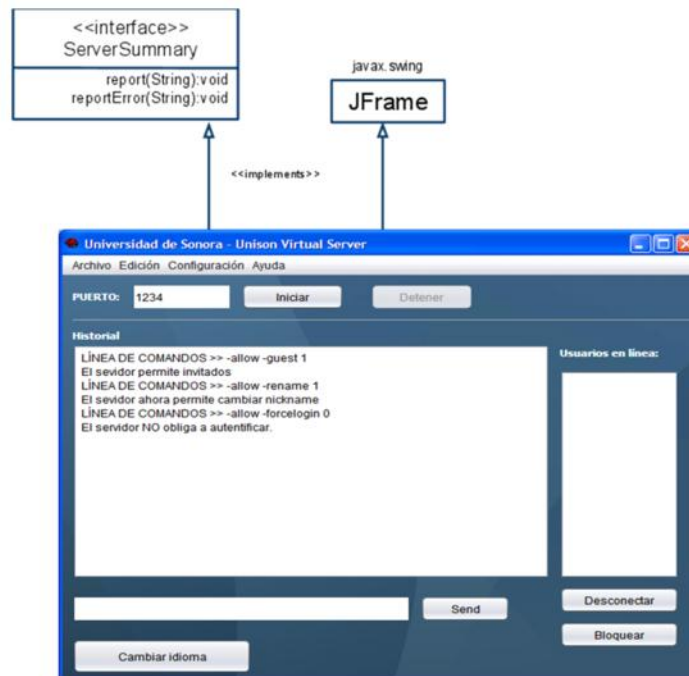


Fig. 8 Vista del Servidor.

Implementa la <<interface>> ServerSummary (ver diagrama de la Fig. 7).

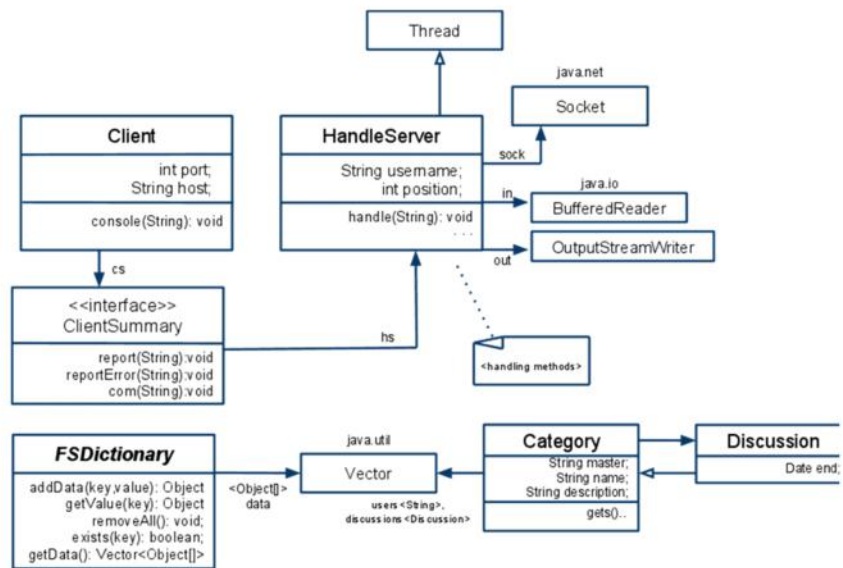


Fig. 9 Diagrama de clases Client



Fig. 10 Vista principal del cliente.

Implementa la <<interface>> ClientSummary (ver diagrama de la Fig. 9)

Interfaz final

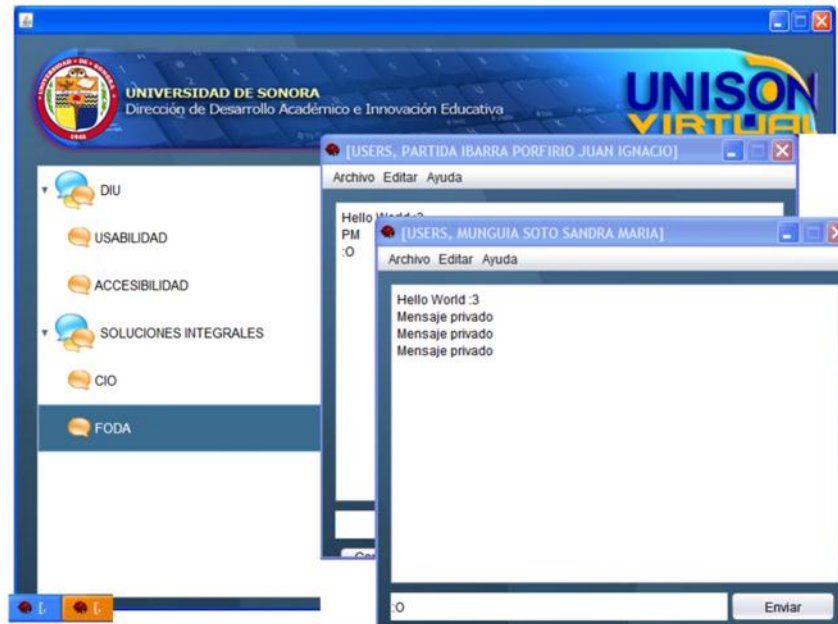


Fig. 11 Vista de las ventanas de discusión.

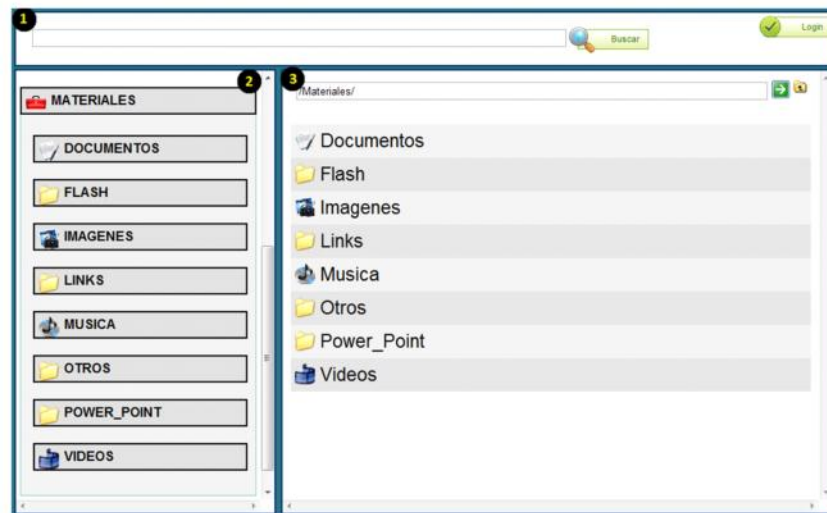


Fig. 12 Interfaz final del repositorio de objetos de aprendizaje

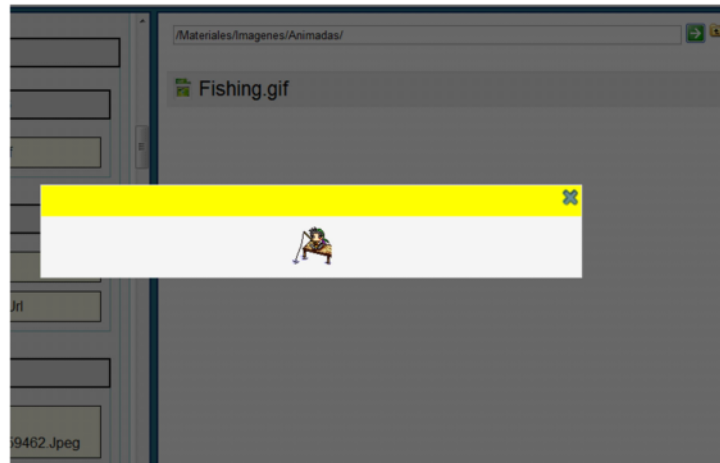


Fig. 13 Interfaz final mostrando una imagen.

CONCLUSIONES Y RECOMENDACIONES

En conclusión, este proyecto me sirvió para experimentar varias áreas laborales, algunas referentes a la carrera en el área que más disfruto de la misma, la programación, otras en un área vital en Ingeniería de Software con tan poca dedicación como lo es el Diseño de Interfaces, no siento que haya habido un aprendizaje (fuera del práctico) que impacte directamente en el perfil profesional como Ingeniero de Software, pero sí hubo mucho aprendizaje por estar tratando con personas directamente, personas que no me veían ahí como a un compañero de trabajo. Detecté muchas oportunidades de mejora, pero es una tristeza que lo que más abundaba en ese ambiente de trabajo era el conformismo. Probablemente el personal tenía demasiado trabajo, pero al menos el tiempo que me tocó ver realmente lo que hacían, lo más productivo que llegaba a haber era trabajo de secretariado sencillo. De tal manera que, mi recomendación es no mandar practicantes directamente a esa área o al menos no tener que depender del personal ahí, la jefa directa de ellos, la subdirectora de DACIE, no siempre está presente, pero al menos ella mostró tener ideas de Innovación y la invitamos a que propusiera sus proyectos en la carrera de manera directa, ya que sus subordinados (la mayoría de ellos), “tienen demasiado que hacer”.

RETROALIMENTACIÓN

En UniVirtual utilizan una metodología de reconversión muy bien elaborada, sin embargo no se cuenta con suficiente personal. Una de las debilidades detectadas es que la jefa del equipo de desarrollo (Subdirectora de DACIE) no conoce o no conocía el alcance de los Ingenieros en Sistemas de Información, de manera que no explota sus habilidades, porque tras ver el proyecto presentado y entrevistarle, tenía muchas ideas, pero no sabía que se podían resolver con Tecnologías de Información, pese a que tiene al menos a dos Ingenieros en Sistemas a su cargo.

Otra problemática detectada en el transcurso de la estancia profesional fue la relación con el asesor asignado de UniVirtual para el proyecto, ya que la información brindada para obtener datos para el Análisis era escasa.

Lo peor que se detecta en el área de desarrollo de UniVirtual, es la falta de interés o bien conflictos personales, ya que en muchas ocasiones fuimos a perder tiempo, ya que dependíamos de una persona “muy ocupada”, y que cuando estaba presente la jefa de UniVirtual, todos resultaban ser muy trabajadores, lo cual es una tristeza.

El diseñador gráfico, nunca nos proporcionó material, por lo que se tuvo que acudir a buscar en internet.

El acceso por FTP nunca pudo funcionar y no se nos permitía usar Escritorio Remoto.

El encargado de solicitar el dominio del sitio tardó más de dos semanas en llevar el formato de solicitud, y sólo lo hizo hasta que comentamos con la jefa de UniVirtual que eso era lo único que nos detenía para presentar.

En general, UniVirtual tiene mucho que mejorar y para ello es necesario que su personal muestre interés en lo que hace.

A lo largo de este documento se habló mucho sobre la dependencia de Moodle y cómo era necesario quitarlo, sin embargo, ese fue el objetivo principal con el que se nos ofreció esta estancia, y curiosamente, nadie del equipo de desarrollo tenía idea de que se quería llegar a ese punto.

FORTALEZAS Y DEBILIDADES

Una de las debilidades que experimenté en la estancia profesional fue el tener que depender de que alguien más sea el responsable de tareas que impactan directamente a mis intereses, en este caso el proyecto del Repositorio de Objetos de Aprendizaje, ya que el asesor asignado de UniVirtual no conocía métodos para darnos acceso al servidor ni tenía tiempo para hacerlo y no teníamos autorización para hacer movimientos de este tipo. Posteriormente, se nos mandó con alguien más que sí podía darnos acceso, pero que tampoco tenía tiempo, de tal manera que nos vimos en la necesidad de hacer presentaciones con servidores externos a univirtual (<http://partidap7.isi.uson.mx/univirtual>).

Al mismo tiempo, pude encontrar fortalezas para seguir trabajando pese a que “había trabas”.

Me sirvió mucho como experiencia, el experimentar lo difícil que es encontrar personas interesadas en hacer lo que tienen que hacer.

Tuve la oportunidad de poner en práctica conocimientos que no había aplicado antes en otras áreas así como la respuesta de otros al ver como se dedicó bastante al diseño de la interfaz.

OPORTUNIDADES DETECTADAS

La cantidad de oportunidades detectadas fue enorme, hay mucho trabajo que hacer en educación a distancia. Principalmente porque pertenece al área de Innovación Educativa y la respuesta del exterior es grande. La subdirectora de DACIE tras exponer este proyecto se mostró muy interesada en dar continuidad y crecimiento a la misión y visión del mismo, reconvertir todas las materias o la mayoría de ellas a modalidad virtual, tras ver el alcance de un sistema hecho en relativamente poco tiempo, le surgieron muchas ideas de mejora.

REFERENCIAS BIBLIOGRÁFICAS Y VIRTUALES

- [1] Bernard H. Boar.
The Art of Strategic Planning for Information Technology, Second Edition.
Publisher: Robert Ipsen
Editor: Theresa Hudson, 2001
- [2] Robert J. Benson, Thomas L. Bugnitz, William B. Walton.
From Bussiness Strategy to IT Action,
John Wiley & Sons, Inc., 2004
- [3] Kenneth E. & Julie E. Kendall
Análisis y Diseño de Sistemas, Sexta Edición
PEARSON EDUCACIÓN, México, 2005
- [4] Toni Granollers i Saltiveri, Jesús Lorés Vidal, José Juan Cañas Delgado.
Diseño de Sistemas Interactivos Centrados en el Usuario
UOC, 2005
- [5] Robert C. Martin.
UML Para Programadores Java.
PEARSON EDUCACIÓN, México, 2004
- [6] Wikimedia Foundation Inc, Junio de 2011
http://en.wikipedia.org/wiki/Software_documentation
- [7] Psicología del color
Como actúan los colores sobre los sentimientos y la razón.
Biblioteca de Arquitectura:
BF789.C7
.H44
26/3075
- [8] Jakob Nielsen's,
How Users Read on the Web
<http://www.useit.com/alertbox/9710a.html>
Alertbox, October 1997.

[9] Wikimedia Foundation,
<http://es.wikipedia.org/wiki/Framework>

[10] Kohana Framework,
<http://kohanaframework.org>

[11] DCMI (Dublin Core Metadata Initiative) Contenido: Título, Palabras clave, Descripción, Fuente, Idioma, Relación, Cobertura.
Propiedad Intelectual: Autor, Editor, Colaboradores, Derechos
Instanciación: Fecha, Tipo de recurso, Formato, Identificación de recurso (id)
http://es.wikipedia.org/wiki/Dublin_Core

ANEXOS

Propuesta

Como propuesta de solución integral, sugerimos el rehacer completamente la plataforma, recreando la forma en que se imparten los cursos, la manera en que se agregan recursos y herramientas, integrar las herramientas propias, gestionar alumnos, maestros y demás roles de usuarios, no es necesariamente un cambio a la interfaz visual, sino al código del proyecto principal. Pero esta vez, rehacerlo utilizando un *Framework*, los cuales son:

- Un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.
- En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado ^[9].

Nuestra propuesta es un framework con el paradigma Modelo - Vista - Controlador o MVC (Model, View, Controller). PHP, el lenguaje de programación utilizado como principal herramienta cuenta con una amplia gama de frameworks. *Kohana* ^[10] es uno de ellos y sigue el paradigma anteriormente mencionado, el cual es muy completo en funciones y fácil de aprender, además de ser uno de los más ligeros y tener una gran escalabilidad.



Fig. 14 Estructura del Framework MVC.

Al utilizar un framework MVC se permite separar en 3 grandes módulos el sistema.

Cuando el usuario haga una solicitud al servidor entra en proceso el framework, el servidor “busca” a la clase controladora de la petición hecha, tal clase realiza consultas, acciones y la programación necesaria para la gestión adecuada, de ser necesario se provee de la clase modelo, la cual se encarga regularmente de las interacciones con nuestro manejador de base de datos y sus asociaciones.

Al final, el controlador *responde* al cliente que hizo tal petición mediante un contenido puramente web denominado vista. De manera que nos permite separar los casos a realizar y encontrar fácilmente las problemáticas, añadir módulos al sistema, gestionar el control de usuarios y optimizar gradualmente nuestro código fuente.

Nos permite apegarnos al paradigma de la orientación a objetos.

Kohana – Breve Guía

Nuestra propuesta es un framework con el paradigma Modelo - Vista - Controlador o MVC (Model, View, Controller). PHP, el lenguaje de programación utilizado como principal herramienta cuenta con una amplia gama de frameworks. *Kohana* es uno de ellos y sigue el paradigma

anteriormente mencionado, el cual es muy completo en funciones y fácil de aprender, además de ser uno de los más ligeros y tener una gran escalabilidad.

Para más detalles sobre éste framework: <http://kohanaframework.org>



Fig. 48 Estructura del Framework MVC.

Al utilizar un framework MVC se permite separar en 3 grandes módulos el sistema.

Cuando el usuario haga una solicitud al servidor entra en proceso el framework, el servidor “busca” a la clase controladora de la petición hecha, tal clase realiza consultas, acciones y la programación necesaria para la gestión adecuada, de ser necesario se provee de la clase modelo, la cual se encarga regularmente de las interacciones con nuestro manejador de base de datos y sus asociaciones. Al final, el controlador *responde* al cliente que hizo tal petición mediante un contenido puramente web denominado vista. De manera que nos permite separar los casos a realizar y encontrar fácilmente las problemáticas, añadir módulos al sistema, gestionar el control de usuarios y optimizar gradualmente nuestro código fuente. Nos permite apegarnos al paradigma de la orientación a objetos.

Por ejemplo

`www.misitio.com/alumno`

hace un llamado al *controlador alumno*, en *Kohana*, se llama al método `index` de la clase `Alumno`.

www.misitio.com/alumno/tarea/4

Con el mismo framework, se llamaría a la clase Alumno, al método *tarea* y se manda como argumento el número 4, es decir, se busca en la sección de controladores la definición

```
public class Alumno extends Controller {  
  
    public          tarea          ($arg){  
    }  
  
}
```

Si el método *tarea*, muestra todas las tareas por default, y la tarea específica en caso de enviarse una, de seguro el código en *tarea* hará una instancia del Modelo para obtener todas las tareas del usuario en cuestión y mostraría la vista que tiene la apariencia adecuada.

```
public tarea ($arg = 0){  
  
    $user = $this->user;  
  
    $modelo = Modelo::instance; // de momento no importa esta sintaxis  
  
    $tareas = $arg == 0 ?  
  
    $modelo->get($user, "tarea"); $modelo->get($user, "tarea", $arg);  
  
        $content = View("tarea.php");  
  
        $content->tareas = $tareas;  
  
  
        echo $content;  
  
}
```

El código anterior puede variar mucho, pero refleja la idea de que el modelo gestiona los datos, y el controlador recibe la petición y contesta con una vista o un conjunto de estas. De manera que en el controlador no aparece jamás código HTML, y en las vistas se muestra código php prácticamente sólo para obtener las variables obtenidas, como el nombre del usuario o los datos a mostrar, como en este caso, que se ha pasado la variable *\$tareas* a la vista de nombre *tarea.php*. Un ejemplo clásico de utilizar una variable, es imprimirla, por lo que la vista podría realizar una función como:

```
<?php foreach($tareas as $tarea) { ?>  
  
<div class="tarea">
```

```
<?php echo $tarea ?>
</div>
<?php } ?>
```

De esa forma, suponiendo que \$tareas es un arreglo, mostrara cada tarea dentro de un div diferente; y en caso de tener problemas al mostrar las tareas, y sabemos que el problema está en el contenido HTML, rápidamente sabremos que tenemos que buscar en la vista *tareas.php* y esta tarea resultará ser sencilla, pues el código es (en este caso) de 5 líneas.

Es necesario modificar los archivos *kohana/index.php* y *kohana/application/bootstrap.php* en la sección de rutas y acciones por defecto para adaptarlas a nuestro sistema.

JQuery – Breve Guía

jQuery es un Framework de JavaScript. Se agrega de la misma manera que cualquier recurso de este lenguaje.

```
<script src="jquery.js">
```

Utiliza funciones que esperan por elementos para llamar a otras funciones, por ejemplo la función “ready” nos sirve para saber si un documento ya ha terminado de ser cargado para entonces inicializar procesos.

Ya sea dentro de un espacio de script o desde otro archivo importado, la sintaxis para saber si el documento está listo es la siguiente:

```
$(document).ready(
    function(){
        //Acción para cuando esté listo.
    }
)
```

```
);
```

Con JQuery se puede llamar a elementos en conjunto o por su id, por ejemplo, `$("#div")` hace referencia a todos los elementos con etiqueta "div" que hayan sido cargados hasta el momento en que se hace llamado a esta función. `$("#sandra")` hace llamado al elemento cuyo ID es "sandra" (.. class = "sandra" ..)

En este framework existen funciones prediseñadas cuyo uso puede resultar común en nuestro sistema, como "hide","show", cuyos parámetros son las cadenas "slow" para hacerlo de manera lenta, o enteros para determinar la duración en milisegundos.

Para ocultar todos los "div" lentamente cuando el documento se cargue:

```
$(document).ready(  
    function(){  
        $("#div").hide("slow");  
    }  
);
```

También se puede acceder al estilo para manipularlo o sólo leerlo.

```
$("#div").css("width", "50") ;
```

```
$("#div").css("width");
```

Para más información revisar la documentación de JQuery en su sitio.

Agregando index.php

En ambos casos, Apache e IIS se requiere tener instalados los módulos de renombrar url, en Apache tal módulo es RewriteEngine y en IIS es URLRewrite. Si en Apache viene instalado, basta con revisar el archivo de configuración y activar (quitar el comentario) en RewriteEngine o Rewrite Module.

Apache

En nuestra carpeta raíz de Kohana necesitamos un archivo llamado “.htaccess” cuyo contenido es el siguiente:

```
# Turn on URL rewriting
RewriteEngine On

# Installation directory
RewriteBase /repositorio/
Options -Indexes

# default page to load
DirectoryIndex /repositorio/index.php/inicio

# ensure there is no /index.php in the address bar
RewriteCond %{THE_REQUEST} ^[A-Z]{3,9}\ /*index\.php\ HTTP/
RewriteRule ^index\.php/(.*) $1 [NS,NC,L,R=301]

# Protect application and system files from being viewed
RewriteRule ^(application|modules|system) - [F,L]

# Rewrite all other URLs to index.php/URL
RewriteRule .* index.php/$0 [PT,L]
```

IIS

En nuestra carpeta raíz de Kohana necesitamos un archivo llamado “web.config” cuyo contenido es el siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
```

```
<system.webServer>
  <defaultDocument>
    <files>
      <add value="index.php" />
    </files>
  </defaultDocument>
  <rewrite>
    <rules>
      <rule name="Kohana" stopProcessing="true">
        <match url="^(.+)$" />
        <conditions logicalGrouping="MatchAll">
          <add input="{REQUEST_FILENAME}"
matchType="IsFile" negate="true" />
          <add input="{REQUEST_FILENAME}"
matchType="IsDirectory" negate="true" />
        </conditions>
        <action type="Rewrite"
url="index.php/{R:1}" />
      </rule>
    </rules>
  </rewrite>
</system.webServer>
</configuration>
```