Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ingeniería en Ciencias y Sistemas

**CREACION DE SISTEMA DE COORDINACION DE PRACTICA FINAL PARA LA ESCUELA DE INGENIERIA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERIA, USAC**

**Boris Aramis Aguilar Rodríguez**

**Walter Omar Vides Rodríguez**

Asesorado por el Ing. Pedro Pablo Hernández

Guatemala, \_\_\_\_ de 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**CREACION DE SISTEMA DE COORDINACION DE PRACTICA FINAL PARA LA ESCUELA DE INGENIERIA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERIA, USAC**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA

FACULTAD DE INGENIERÍA

POR

**BORIS ARAMIS AGUILAR RODRIGUEZ**

**WALTER OMAR VIDES RODRIGUEZ**

ASESORADO POR EL ING. PEDRO PABLO HERNÁNDEZ

AL CONFERÍRSELES EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, \_\_\_\_ DE 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO Ing. Murphy Olympo Paiz Recinos

VOCAL I Ing. Alfredo Enrique Beber Aceituno

VOCAL II Ing. Pedro Antonio Aguilar Polanco

VOCAL III Ing. Miguel Ángel Dávila Calderón

VOCAL IV Br. Juan Carlos Molina Jiménez

VOCAL V Br. Mario Maldonado Muralles

SECRETARIO Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**CREACION DE SISTEMA DE COORDINACION DE PRACTICA FINAL PARA LA ESCUELA DE INGENIERIA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERIA, USAC**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha septiembre de 2013.

**Boris Aramis Aguilar Rodríguez**

**Walter Omar Vides Rodríguez**

Índice general

ÍNDICE DE ILUSTRACIONES III

LISTA DE SÍMBOLOS IV

GLOSARIO V

RESUMEN VI

OBJETIVOS VII

INTRODUCCIÓN VIII

1. MARCO TEÓRICO 1

1.1. Web2py 1

1.1.1. Modelo 1

1.1.2. Vista 1

1.1.3. Controlador 2

1.2. Python 2

1.3. LMS 2

1.3.1. Chamilo 3

1.4. jQuery 3

1.5. MySQL 4

2. FASE DE INVESTIGACIÓN 5

2.1. Antecedentes 5

2.2. Descripción de las necesidades 5

2.2.1. Control de Áreas del proyecto DTT 6

2.2.2. Control de reportes 6

2.2.3. Control de estudiantes 7

2.2.4. Control de proyectos 7

2.2.5. Integración con Chamilo 7

2.2.5.1. Rol de catedrático y rol de estudiante 8

2.2.5.2. Rol de director de escuela 8

2.2.6. Limitantes de la Integración con chamilo 8

2.2.7. Control y seguimiento de proyectos 9

2.2.7.1. Reporte de resultados de actividad de curso 10

2.2.7.2. Bitácora de actividades efectuadas durante el mes 10

2.2.7.3. Reportes almacenados en archivos 10

2.2.7.4. Tareas que se califican como cumplida o no cumplida 11

3. FASE TÉCNICO PROFESIONAL 12

3.1. Configuracion de servidor de producción 12

4. FASE ENSEÑANZA APRENDIZAJE 12

4.1. Roles y accessos 12

conclusiones 13

Recomendaciones 14

bibliografía 15

ÍNDICE DE ILUSTRACIONES

**FIGURAS**

**TABLAS**

LISTA DE SÍMBOLOS

**Símbolo Significado**

GLOSARIO

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

RESUMEN

El presente trabajo de EPS consiste en la creación de un sistema informático que lleve el control de los estudiantes de práctica final de la escuela de Ingeniería en Ciencias y Sistemas de la Universidad de San Carlos de Guatemala.

Se crean las herramientas necesarias para el control de los estudiantes que están realizando la práctica final de forma independiente de la institución donde la realizan pero bajo las distintas modalidades necesarias en la Escuela de Ciencias y Sistemas.

El objetivo de estas herramientas es permitir que los estudiantes, catedráticos y los encargados de llevar el control de la práctica final puedan realizarlo de manera efectiva afrontando el crecimiento estudiantil.

Otro punto a cubrir en el presente trabajo, es la integración con un sistema existente de *e-learning*, con el propósito de que exista menos duplicidad en la información de los distintos sistemas existentes en la escuela.

Por último, se creó la documentación de los procesos del sistema y sus distintas herramientas. Esto con la finalidad de orientar a los usuarios administradores, estudiantes y supervisores de la práctica final en el uso de las herramientas.

OBJETIVOS

**General**

Crear un sistema de Control de Practica Final para la escuela de Ingeniería en Ciencias y Sistemas que pueda administrar los distintos procesos de práctica final que se llevan en la escuela logrando que sean más eficientes, tanto para los administradores como para los catedráticos y practicantes finales.

**Específicos**

1. Facilitar el control de la información que se obtiene al realizar la práctica final en la Escuela de Ciencias y Sistemas.
2. Aumentar la amplitud de áreas en las cuales se automatiza el proceso de práctica final en la Escuela de Ciencias y Sistemas.
3. Permitir un flujo centralizado de información y datos referentes a la práctica final, dando la oportunidad de conocer el estado en cualquier momento de las actividades que forman parte de la práctica final dentro de la Escuela de Ciencias y Sistemas.
4. Automatizar el proceso de entrega de informes por parte de los estudiantes de práctica final de la Escuela de Ciencias y Sistemas..

INTRODUCCIÓN

La automatización de procesos es muy importante, cuando automatizamos una actividad podemos disponer de mayor control de la misma y satisfacer mayores demandas en esa actividad sin tener que recurrir a contratar personal adicional, optimizando recursos y mejorando la calidad y velocidad del flujo de información.

La Escuela de Ciencias y Sistemas tiene como cargo velar por el cumplimiento de la práctica final que se desarrolla por todos los estudiantes que finalizan el proceso de práctica como parte de su pensum de estudios. Esto implica que se vela por el cumplimiento de todas estas labores a través de informes que indican el estado de la práctica por cada estudiante y área a la que pertenece. Durante años anteriores se ha definido una estructura que compone la escuela de sistemas, esta estructura se compone de distintas áreas que proveen y cumplen cierta necesidad y expectativa de la escuela. Los estudiantes de práctica final ejecutan proyectos que pertenecen a cada una de esas áreas.

Anteriormente existía un sistema que cubría la automatización de control de reportes de práctica final de la escuela de ciencias y sistemas para los practicantes que se desempeñan en el área de tutor académico. Sin embargo, debido a la existencia y requerimiento de automatizar el proceso para todas las áreas se requiere de la creación de un nuevo sistema que realice esas tareas.

Debido a estos requerimientos tecnológicos en la automatización de procesos se observa la necesidad de crear un nuevo sistema que provea la flexibilidad necesaria para cumplir con los requerimientos de flexibilidad en el sistema para soportar las distintas áreas que componen la escuela de sistemas y poder llevar el control de los practicantes que se desempeñan en las mismas.

Se inicia con una investigación preliminar con el objetivo de elegir las mejores herramientas y tecnologías a utilizar para la creación de este nuevo sistema que busca mantener automatizados varios procesos ligados a la práctica final de los estudiantes y por ende también a varios involucrados en el mismo, el cual incluye catedráticos y encargados de la escuela de ciencias y sistemas..

Luego de la selección preliminar de lo que se va a utilizar para el proyecto, se hace una investigación sobre la forma en que funcionan las herramientas y los componentes para cumplir con los objetivos planteados.

Por último, se implementa la solución se crea la documentación para la utilización de la solución por las partes que utilizaran la misma.

1. MARCO TEÓRICO
   1. Web2py

Web2py es un marco de trabajo (*framework)* de desarrollo ágil para aplicaciones web, que define una metodología de trabajo, herramientas y estándares a seguir para facilitar la creación y mantenimiento de aplicaciones; está basado en el lenguaje de programación python.

Basa su configuración en la arquitectura Modelo Vista Controlador, que consiste en separar los componentes en tres conceptos, el aspecto visual, la lógica del negocio y el almacenamiento de la información, la ventaja principal de utilizar este tipo de configuración, es contar con la posibilidad de modificar uno u otro aspecto de la aplicación sin afectar directamente el desempeño de otra área o el trabajo invertido en otra de las dos áreas.

* + 1. Modelo

El modelo es la herramienta y convenciones de la arquitectura establecidos para definir el cómo se almacena, maneja y accede a la información.

* + 1. Vista

La vista está constituida por la herramienta y lógica empleada para mostrar al usuario de manera gráfica la información, es también medio de interacción entre el usuario y la aplicación.

* + 1. Controlador

El controlador es el componente encargado de recibir las solicitudes de las interacciones entre el usuario y la aplicación y efectuar tareas basándose en la información que esta interacción produce, tomar decisiones, procesar información y devolver una respuesta, que es interceptada por el componente encargado de la vista quien a su vez lo mostrara al usuario para iniciar un nuevo flujo de comunicación que recae en la lógica del controlador.

* 1. Python

Python es un lenguaje de programación interpretado, que está orientado a ser un lenguaje fácil de mantener y entender haciendo uso de sintaxis simple y limpia, cuenta con soporte amplio de varios paradigmas o técnicas de programación.

* 1. LMS

Su nombre es derivado de sus siglas en inglés que se traducen al español como Sistema de gestión de aprendizaje. Se usa para administrar, distribuir y controlar las actividades académicas de forma no presencial. Entre sus funciones están:

* Gestión de usuarios
* Gestión de recursos
* Administración de accesos
* Control y seguimiento del proceso de aprendizaje
* Evaluaciones
* Generar informes
* Servicios de comunicación ( foros, video conferencias).

Algunos ejemplos de un LMS se mencionan a continuación.

* + 1. Chamilo

Es una solución web de código abierto de *e-learning* y de gestión de contenidos, que tiene como propósito mejorar el acceso a la educación y comunicación con un costo reducido. Fue lanzado el 18 de enero de 2005 por una parte de la comunidad de Dokeos, después de un gran descontento sobre la política de comunicación entre esta comunidad, por esto Chamilo es considerado un *fork* de Dokeos. Chamilo está compuesto por diferentes elementos como un servidor web, un manejador de base de datos y un sistema de archivos, fue creada para ser ejecutada sobre la plataforma LAMP.

Entre las características importantes de esta herramienta, podemos mencionar las siguientes:

* Creación de contenidos educativos
* Fácil instalación en soluciones de web *hosting*
* Seguimiento de los resultados de los usuarios, que permiten mejorar la metodología
* Interfaz limpia, dejando que el usuario se centre en el aprendizaje
* Muchas herramientas que permiten todos los tipos de aprendizaje
* Amplia gestión de documentos
* Licencia GNU/GPLv2
  1. jQuery

*jQuery* es una librería basada en JavaScript que sirve para simplificar las operaciones de sobre el árbol DOM que contiene los elementos que componen las páginas web en los navegadores, también cuenta con una variedad de funciones que simplifican operaciones visuales, de comunicación entre sitios y comunicación entre el cliente y el servidor.

Fue creada por John Resig y fue presentada el 14 de enero de 2006, es también la librería de JavaScript más utilizada, entre sus objetivos se encuentra simplificar el uso de la técnica AJAX, es libre y de código abierto, el uso de la librería agiliza y simplifica tareas y operaciones que de otra manera consumirían mucho tiempo y espacio.

* 1. MySQL

Sistema de administración de base de datos ( DBMS) para la gestión de bases de datos relacionales, que fue escrito en C y C++, y destaca por su gran adaptación a diferentes entornos de desarrollo. Permite la interacción con lenguajes de programación como PHP, Perl y Java. Además es compatible también con distintos sistemas operativos.

MySQL es un servidor de base de datos multi-usuario. Es robusto para la ejecución de instrucciones en paralelo. Múltiples usuarios distribuidos en cualquier red, pueden ejecutar distintas tareas sobre las bases de datos dentro de un mismo servidor. Utiliza el lenguaje SQL estándar para la ejecución de instrucciones.

1. FASE DE INVESTIGACIÓN
   1. Antecedentes

La Escuela de Ciencias y Sistemas cuenta con un sistema que lleva el control de estudiantes de práctica final en modalidad de auxiliatura llamado caecys. Este sistema permite llevar el control de las distintas actividades que realiza un auxiliar. También se posee de otro sistema de *e-learning* llamado Chamilo, el cual posee la información de los alumnos y catedráticos de la Escuela de Ciencias y Sistemas.

Estos dos sistemas son la parte fundamental del control del proceso y progreso académico de los estudiantes de la Escuela de Ciencias y Sistemas.

Sin embargo, estos 2 sistemas trabajan de forma independiente y la información muchas veces es duplicada entre ellos, además que durante el último año se realizó un cambio en la modalidad de práctica final que distribuye a los alumnos practicantes finales en distintas áreas que no son únicamente el realizar labores de auxiliares en los cursos de la escuela.

* 1. Descripción de las necesidades

Bajo los nuevos requerimientos de labores de practicantes finales era necesario el realizar un sistema que ayudara a llevar el control de ello, ya que el sistema actual no contempla las distintas modalidades práctica final.

Es necesario también que ese control sea congruente en el proceso académico y por ello se necesita ser integrado con la plataforma de *e-learning* de la escuela.

Los estudiantes de práctica final son asignados como participes del programa DTT y cada sección del programa tiene distintos requerimientos que cumplir por parte del practicante final.

El progreso de práctica final de cada estudiante debe ser reportado a la escuela de Ciencias y Sistemas, integrando como parte de su contenido la participación dentro del proyecto al cual están asignados en forma de reportes, el objetivo del sistema es automatizar ese proceso.

Adicionalmente hay tareas especificas por área sobre las cuales se desea llevar control, todas estas tareas se definen y existen algunas que se repiten cada semestre de forma constante.

* + 1. Control de Áreas del proyecto DTT

El programa DTT de la facultad de ingeniería contempla dentro de sus actividades promover y funcionar como intermediario entre los estudiantes de la facultad y las áreas laboral, de investigación, y otros proyectos de la universidad, por lo tanto debe también gestionar actividades de reclutamiento de estudiantes de la escuela de ciencias y sistemas, así como llevar control de las actividades, rendimiento y tareas que la ejecución de un proyecto relacionado con DTT requiera.

El proyecto DTT se divide en 6 áreas, que deben ser gestionadas dentro de la administración del sistema a crear, las cuales son:

* Tutor académico
* Tutor de comunicación
* Tutor de desarrollo
* Tutor de innovación
* Tutor de investigación
* Tutor de infraestructura
  + 1. Control de reportes

El sistema de control de progreso y calificación de actividades realizadas por los estudiantes integrantes del programa es realizado por medio de la entrega de reportes que son revisados y calificados por los estudiantes o catedráticos encargados de los estudiantes que realizan los proyectos en las áreas especificadas, cuyo rol es el de tutor y puede aprobar o reprobar a un estudiante, el sistema de reportes actual no cuenta con un sistema de calificación basado en notas, esta es una de las mejoras a implementar en el nuevo sistema.

* + 1. Control de estudiantes

El control de estudiantes permite organizar y presentar de manera simple la métricas y reportes a los usuarios con roles administrativos de la aplicación, así como condensar la información y filtrarla para soportar la toma de decisiones relacionadas con cambios en la red de estudios y cambios directamente relacionados con las áreas y proyectos del programa DTT.

* + 1. Control de proyectos

Directamente relacionados con las áreas del programa DTT se encuentran los proyectos en los que los estudiantes se involucran haciendo actividades según la naturaleza y área del proyecto, en las que aplican conocimientos adquiridos durante la carrera, consolidándolos así con aplicación real de los mismos.

* + 1. Integración con Chamilo

La integración con la actual universidad virtual de la escuela de ciencias y sistemas con el nuevo sistema de control de proyectos finales de la escuela de ciencias y sistemas es uno de los requerimientos que fueron solicitados al momento de proponer la reingeniería del sistema, se determinó que los roles a cubrir dentro del alcance de la integración son los siguientes:

* Rol de practicante final
* Rol de catedrático
* Rol de administrador de sistema
* Rol de DSI

Dado que los roles catedrático y practicante final cuentan con un comportamiento similar dentro de la aplicación, serán explicados de manera unificada a continuación en del detalle de alcances y comportamientos esperados de la integración de cada rol con el sistema chamilo ya existente.

* + - 1. Rol de catedrático y rol de practicante final

La integración con los roles de estudiante y de catedrático dentro de chamilo, requiere que el estudiante cuente con acceso como el de un curso dentro de su asignación, que muestre contenido relacionado con el proyecto y que enlace la pantalla de cursos con el panel de control.

* + - 1. Rol de administrador de sistema

El rol de administrador de sistema requiere principalmente la posibilidad de generar reportes relacionados con la información que se genera de las actividades de los usuarios en los proyectos de las áreas del programa DTT, estos reportes están constituidos de información condensada, métricas y gráficas basadas en la información que puede ser medida y representada numéricamente. Adicionalmente este rol presenta toda la gestión de información que forma parte del sistema.

* + - 1. Rol de DSI

Los encargados de llevar el control de aspectos de cumplimiento para entregas y horas de soporte por parte de estudiantes que pertenecen a la sección de Tutor Académico son los practicantes que pertenecen al departamento de soporte informático (DSI). Este departamento necesita acceso al sistema para verificar la validez de asistencia a el área correspondiente por parte de los practicantes finales que pertenecen a área de Tutor Académico.

* + 1. Limitantes de la Integración con chamilo

La integración ideal entre los sistemas es en la que se codifica dentro del sistema chamilo un grupo de módulos destinados a manejar y presentar la integración, esto debería ser realizado usando las herramientas de integración de chamilo destinadas a extender el código y funcionalidad del mismo, del proceso de investigación previo se determinó que las herramientas de desarrollo de la versión de chamilo que se tiene en uso en la facultad nunca fueron terminadas por los desarrolladores.

Dado que no se cuenta con las herramientas adecuadas de chamilo para efectuar la integración de manera correcta, se utilizará la herramienta *jQuery* que proveerá para la integración la comunicación entre ambos sistemas, los componentes visuales y transiciones entre sistemas.

La librería *jQuery* proveerá de las funciones necesarias para inyectar dentro del contenido del árbol DOM de la página de chamilo el contenido adicional que será determinado según el rol del usuario.

* + 1. Control y seguimiento de proyectos

El programa DTT está dividido en 5 áreas que se subdividen en proyectos, el control de dichos proyectos depende directamente del área a la que pertenecen, y están relacionados con uno de dos tipos de limitantes de tiempo, las limitantes pueden ser:

* Ciclo de estudio, que representa el semestre de un año
* Rango de tiempo, que generalmente es de una semana.

Los reportes pueden estar presentes en todas o algunas de las áreas y pueden contar con métricas y entregables, los entregables se categorizan según su naturaleza en los siguientes:

* Reporte de resultados de actividad de curso
* Bitácora de actividades efectuadas durante el mes
* Reportes almacenados en archivos
* Tareas que se califican como cumplida o no cumplida
  + - 1. Reporte de resultados de actividad de curso

Los reportes de resultados de una actividad del curso son alimentados por los resultados de evaluaciones realizadas durante actividades, estos resultados son información medible, o métricas, algunas de estas suelen ser:

* Cantidad de estudiantes participando en la actividad
* Número de estudiantes que aprobaron la actividad
* Número de estudiantes que reprobaron la actividad
* Nota promedio
* Desviación estándar
  + - 1. Bitácora de actividades efectuadas durante el mes

Los reportes que no pueden ser registrados cuantificados o representados de manera medible, serán registrados en el sistema como actividades realizadas o anomalías, según sea el caso, en algunos casos, estas actividades estar relacionados con reportes de métricas de una actividad, siendo la entrada en la bitácora la descripción de la actividad y un reporte de resultado de actividad de curso puede contener las métricas obtenidas durante dicha actividad.

* + - 1. Tareas a entregar como archivo

Las tareas a entregar como archivos pueden ser actividades emergentes, o bien permanentes, ejemplos de actividades emergentes pueden por ejemplo ser una visita técnica de mantenimiento de a alguna institución, mientras que uno permanente puede ser un programa de curso u otro documento relacionado con un proyecto, que requiera ser generado todos los semestres.

* + - 1. Tareas que se califican como cumplida o no cumplida

Este tipo de tareas generalmente suelen ser emergentes, y servirán para determinar si un usuario participó o no en una actividad y el administrador de sistema puede definir que sean marcadas como cumplidas o incumplidas por parte del DSI, por parte del encargado del proyecto al que pertenece la tarea o por parte de su rol administrador.

* + - 1. Tareas que se califican con nota

Este tipo de tareas generalmente suelen ser emergentes, y servirán para determinar si un usuario participó o no en una actividad y el grado de participación que tuvo en la misma. El administrador de sistema puede definir que sean marcadas como cumplidas o incumplidas por parte del DSI, por parte del encargado del proyecto al que pertenece la tarea o por parte de su rol administrador.

1. FASE TÉCNICO PROFESIONAL
   1. Implementación de Funcionalidad para Control de Práctica Final en Escuela de Ciencias y Sistemas

La implementación del sistema implica la creación de diversas partes, los detalles de la implementación así como su comportamiento se describen en las siguientes sub secciones.

* + 1. Administración de Áreas de DTT

El departamento DTT consta de 6 áreas sobre las cuales se asignan los distintos proyectos de práctica final. Estas áreas se pueden modificar en el sistema y a cada una se le asigna automáticamente un código. Los proyectos dependen de un área específica.

* + - 1. Administración de proyectos

Un proyecto se lleva a cabo como parte de un área de DTT, el administrador del sistema puede añadir, crear y modificar cualquier proyecto conforme sea necesario.

* + 1. Administración de Usuarios

Los usuarios tienen roles, los roles existentes son: Administrador, DSI, Practicante Final y Catedrático o supervisor de proyecto. El administrador puede realizar todas las tareas de administración y consultad de datos que requiera.

El administrador puede crear Usuarios nuevos ingresando los datos solicitados, posteriormente le asigna uno o varios roles al usuario recién creado. Al tener un rol el usuario ya puede pertenecer a un proyecto, dependiendo el rol del usuario será el conjunto de operaciones que puede realizar dentro del sistema.

* + 1. Activación de práctica final y asignación a proyecto

El proceso de activar la práctica final para los estudiantes se efectúa de forma semestral y se realiza en fechas determinadas, en esas fechas se realiza una carga en el sistema con los alumnos que son ahora practicantes finales.

* + - 1. Carga de Practicante Final por medio de archivo

El formato para cargar los alumnos que se asignan práctica final es CSV y los campos necesarios son Correlativo, Carnet, Nombre, Código de Proyecto, Duración, Ad Honorem, email y Horas por cumplir. De estos campos, se pueden omitir el Nombre y las horas por cumplir; si se omiten se buscaran automáticamente en los datos de la UV, si esto no es posible se le solicitaran sus datos al practicante al ingresar al sistema por primera vez.

* + - 1. Primer Ingreso de practicante final

Posterior a la carga del usuario practicante final, este puede ingresar al sistema solicitando una contraseña; esta contraseña llegará al usuario por medio de correo electrónico y junto a ella una URL única que le permitirá colocar una contraseña deseada y finalizar ingresando los datos que requiere cada usuario para continuar. Los datos que se requieren son:

* Nombres Completos
* Números de Teléfono
* Correo Electrónico
* Estado laboral y dirección
  + - 1. Entrega e Ingreso de Reportes

El administrador del sistema debe habilitar la creación de reportes a los practicantes finales, los reportes se entregan en fechas específicas y se habilitan durante un periodo de tiempo. Los reportes se dividen en Finales y Parciales, la diferencia radica en la información que se ingresa en cada reporte. Adicionalmente los reportes independientemente si son finales o parciales obligan al estudiante a ingresar ciertos segmentos de reporte, dependiendo del área a la que pertenece el proyecto que les es asignado.

* + - * 1. Reporte tipo Final

El reporte tipo final afecta específicamente el área de práctica final que corresponde a Tutor académico, ningún otra área se ve afectada por este tipo de restricción. Los reportes tipo final del área de tutor académico deben contener el resumen de deserción y asistencia del semestre así como notas finales del curso del cual el practicante final es tutor académico.

* + - * 1. Reporte tipo Parcial

Todos los reportes que no se indican como finales se consideran parciales, estos se entregan en los periodos que indica el administrador del sistema y aplican a todas las áreas de práctica final. Dependiendo el área de práctica a la que pertenezca una asignación de práctica final es que se requiere una cierta cantidad de segmentos de reporte distintos.

* + - * 1. Segmentos de Reporte Final

Cada área de práctica final que pertenece al proyecto DTT puede poseer una distinta necesidad de información a indicar en cada reporte, cada una de ellas se denomina segmento, los segmentos obligatorios los define el administrador del sistema y cada segmento tiene un propósito en específico a cumplir dependiendo el área de práctica final.

* + - 1. Entrega e Ingreso de Requisitos

Durante el curso de la práctica final el practicante debe llenar varios requisitos solicitados como parte del proceso. Estos requisitos los ingresa el administrador del sistema y se asigna una fecha límite para la entrega de los mismos. Como parte del cumplimiento de la práctica final se presenta como obligatorio estos requisitos para considerar aprobada la misma. Existen requisitos de tipo Archivo, Actividad, Actividad con Nota y Horario.

Los requisitos de Archivos son documentos que se solicitan al practicante como programas de curso y otros que debe presentar.

Las actividades son el cumplimiento de alguna actividad que debe realizar un practicante.

Los requisitos de tipo Actividad con Nota refieren a una actividad que posee un valor de aprobación mínimo, por ejemplo la asistencia como parte de DSI para los practicantes que pertenecen al área tutor académico y deben cumplir un 80% de esa asistencia.

Horario refiere a un requisito de ingresar al sistema un horario que indique algún tipo de información solicitada por el administrador del mismo.

* 1. Configuración de servidor de producción

El servidor donde se instalan los componentes necesarios para este EPS consta de un sistema operativo basado en RHEL 5, específicamente es la versión Centos 5.8 con la última actualización a la fecha (Abril del 2014). Por lo que se identificó la necesidad de instalación de ciertos componentes:

* Apache Versión 2, disponible en repositorios RHEL5 predeterminados.
* Python Versión 2.7 o superior. Debido a que los repositorios de RHEL5 no cuentan con una versión superior a la 2.4 se necesita la actualización a esta versión por parte del administrador del sistema. Esto implica la instalación de los siguientes paquetes: zlib-devel, gcc, python-setuptools, readline-devel; todos estos disponibles en el repositorio RHEL5 predeterminado.
* Instalación de módulos para apache y python, específicamente openssh-server que se encuentra en los repositorios RHEL5 predeterminados y mod-wsgi que se debe compilar para la versión específica de apache y python a utilizar (en este caso python 2.7 y apache 2).
  + 1. Instalación de Apache

El servidor Apache es el que responde a las solicitudes de los usuarios, existe una variedad amplia de servidores de contenido como Apache, siendo este el más comúnmente utilizado, para instalar apache en el servidor es necesario contar con permisos de administrador en el equipo a utilizar, en Centos 5 basta con ejecutar las líneas que se observan en la figura [numero\_figura]

1. Instalación de Apache en el servidor

yum –y install http2

chkconfig httpd –add

yum install mod\_wsgi

yum install openssh-server

mkdir /etc/httpd/ssl

yum install mod\_ssl

yum install httpd-devel

Fuente: elaboración propia.

* + 1. Instalación de Python

La instalación de Python es únicamente necesaria en el caso que la versión de Centos sea la versión 5, las versión de Centos 6 o superiores ya cuentan de manera predeterminada con la versión necesaria de Python para el correcto funcionamiento de las herramientas de las que depende el proyecto, en el caso en que se cuente con una versión de Centos superior o igual a la 6 con Python 2.6 o superior, omitir los siguientes pasos, en caso de contar con Centos 5 y versión inferior de Python a la 2.6 ejecutar los comandos de la figura [numero\_figura]

1. Instalación de Python 2.6 en el servidor

yum install -y readline-devel python-setuptools gcc zlib-devel

PATH=~/opt/bin:$PATH

wget http://www.python.org/ftp/python/2.7.5/Python-2.7.5.tgz

tar –xvzf Python-2.7.5.tgz

cd Python-2.7.5.tgz

./configure

make altinstall

Fuente: elaboración propia.

* + - 1. Configuración de módulos de Apache para Python 2.7

Es necesario configurar los módulos requeridos en el archivo de configuración de apache como se muestra en la figura [numero\_figura]

Los módulos configurados en los comandos de la figura [numero\_figura] son necesarios para el funcionamiento de Web2py.

1. Configuración de módulos de Apache

nano /etc/httpd/conf/httpd.conf

**Agregar la siguiente línea en este archivo**

LoadModule ssl\_module modules/mod\_ssl.so

wget https://modwsgi.googlecode.com/files/mod\_wsgi-3.4.tar.gz

tar -xzf mod\_wsgi-3.4.tar.gz

cd mod\_wsgi-3.4

./configure --with-python=/opt/bin/python2.7

make

make install

Fuente: elaboración propia.

* + 1. Instalación de Web2py

La herramienta de software para el funcionamiento de la aplicación requiere que se configure el entorno del servidor objetivo con los directorios, paquetes y configuraciones de apache que se muestran en los siguientes pasos

* + - 1. Instalación de Web2py desde la fuente oficial

Web2py puede ser instalado desde varios recursos y de diferentes maneras, la instalación sugerida es por medio de obtener el archivo compreso desde el sitio oficial, tal como se puede apreciar en la figura [numero\_figura].

1. Configuración de entorno para Web2py

mkdir /var/www/web2py

cd /var/www/web2py

wget http://web2py.com/examples/static/web2py\_src.zip

sudo unzip web2py\_src.zip

chown -R apache:apache /var/www/web2py/web2py

mv /var/www/web2py/web2py/handlers/wsgihandler.py /var/www/web2py/web2py/

Fuente: elaboración propia.

* + - 1. Configuración de Apache para servir la aplicación

Para que el servidor pueda responder correctamente a las solicitudes de los clientes se debe crear un archivo de configuración que agrega el texto mostrado en la figura [numero\_figura]

1. Configuración de rutas de Apache

nano /etc/httpd/conf.d/web2py.conf

WSGISocketPrefix /var/run/wsgi

<VirtualHost \*:80>

ServerName v2.caecnoys.org

WSGIDaemonProcess web2py user=apache group=apache displayname=%{

GROUP}

WSGIProcessGroup web2py

WSGIScriptAlias / /var/www/web2py/web2py/wsgihandler.py

<Directory /var/www/web2py/web2py>

AllowOverride None

Order Allow,Deny

Deny from all

<Files wsgihandler.py>

Allow from all

</Files>

</Directory>

AliasMatch ^/([^/]+)/static/(.\*) var/www/web2py/web2py/applications/$1/static/$2

<Directory /var/www/web2py/web2py/applications/\*/static/>

Order Allow,Deny

Allow from all

</Directory>

<Location /admin>

Deny from all

</Location>

<LocationMatch ^/([^/]+)/appadmin>

Deny from all

</LocationMatch>

CustomLog /var/www/web2py/logs/access.log common

ErrorLog /var/www/web2py/logs/error.log

</VIrtualHost>

Fuente: elaboración propia.

* + - 1. Configuración de directorios de errores y accesos

El marco de trabajo requiere directorios donde almacenar la bitácora de errores y que estos directorios tengan los permisos correctos, dichos permisos permitirá al servidor Apache escribir y sobre escribir la bitácora, como lo muestra la figura [numero\_figura].

1. Directorios y permisos

mkdir /var/www/web2py/logs

chown -R apache:apache /var/www/web2py/logs

Fuente: elaboración propia.

* + 1. Instalación de Base de Datos

Con relación a la persistencia de información, Web2py requiere al menos un motor de base de datos que administración y almacenamiento de datos, para crecimiento futuro y robustez de la aplicación es necesario que sean configurados como mínimo Mysql y Sqlite, los mismos pueden ser instalados usando los comandos de la figura [numero\_figura]

1. Configuracion de Mysql y Sqlite

yum install sqlite sqlite-devel

cd ~

wget https://pysqlite.googlecode.com/files/pysqlite-2.6.3.tar.gz

tar -xzf pysqlite-2.6.3.tar.gz

cd pysqlite-2.6.3

python2.7 setup.py install

wget http://goo.gl/NZ9InB

gunzip mysql-connector-python-1.0.12.tar.gz

tar xf mysql-connector-python-1.0.12.tar

cd mysql-connector-python-1.0.12

python2.7 setup.py install

/etc/init.d/httpd restart

Fuente: elaboración propia.

* + 1. Integración con Chamilo

El proceso de integración de la aplicación CPFECYS con otras herramientas es genérico y por lo tanto puede ser aplicado a cualquier herramienta que haga uso de jQuery, si la herramienta no cuenta con la librería jQuery, la misma puede ser agregada ser agregada al proyecto.

Los pasos a seguir para efectuar la integración con Chamilo, son los siguientes:

* Modificar el archivo de jQuery en el cual se cargan los cursos, el cual es se llama home\_ajax.js
* Agregar el archivo generador de contraseñas de autenticación entre sitios, llamado session\_handler

Esta integración se puede efectuar en los dos pasos anteriores de manera a cualquier otro sistema, la diferencia será únicamente el archivo home\_ajax.js, en otros casos puede ser llamado de otra manera o bien ser un archivo diferente al cual se adjunten las líneas de código que se verán en la figura [figura\_numero]

1. Configuracion de Mysql y Sqlite

$(document).ready(function() {

function getBaseUrl(){

pathArr = window.location.href.split('/');

prot = pathArr[0];

host = pathArr[2];

return prot + '//' + host;

}

var baseUrl = getBaseUrl();

$.get( "session\_handler.php", function(response) {

response = eval( '(' + response + ')')

var session\_id = response.uid;

$.ajax({

type: 'POST',

url: baseUrl + '/[dominio\_ejemplo]/cpfecys/abstract/user\_active.json',

data: {

'uid': session\_id

},

success: function(msg){

if(msg.success){

$('#course\_type\_course\_list\_renderer\_0').append(

'<div class="block user\_category\_block" id="course\_user\_category\_0">

<div class="description user\_category\_description">

<ul style="display: block;"><div style="float:left;">

<li style="list-style: none; margin-bottom: 5px; list-style-image: url(http://162.243.16.29/chamilo5/user/resources/images/aqua//logo/16.png);">

<a id="aux\_click" style="top: -2px; position: relative;" href= "'+baseUrl+'/dtt/cpfecys/abstract/oauth\_login?token=' +response.token+'">AUXILIATURA</a>

<br></li></div>

<divstyle="float:right; padding-right: 20px;"></div><div style="clear: both;"></div></ul></div></div>');

}

}

});

});

});

Fuente: elaboración propia.

El objetivo del archivo session\_handler es acceder a la base de datos de usuarios de la herramienta con la que se desea integrar CPFECYS y generar basándose en el identificador único del usuario un toquen sirve para comparar con el toquen generado en la herramienta CPFECYS, de esta manera se puede determinar comparando la igualdad de ambos si un usuario en la herramienta a integrar cuenta con un usuario activo en CPFECYS, el contenido del archivo session\_handler es el que se muestra en la imagen [imagen\_numero]

1. Contenido del archivo session\_handler

<?php

session\_name('11284b891395526f1b5c26d78dc798f4');

session\_start();

// store session data

//$\_SESSION['views']=1;

?>

<?php

//retrieve session data

//echo $\_SESSION['\_uid'];

//$\_SESSION['\_uid']=2;

if (!isset($\_SESSION['\_uid'])) return;

$mysqli = new mysqli("localhost", "root", "", "chamilo6");

if ($mysqli->connect\_errno) {

echo "Falló la conexión con MySQL: (" . $mysqli->connect\_errno . ") "

. $mysqli->connect\_error;

}

$carnet = NULL;

/\* crear una sentencia preparada \*/

if ($stmt = $mysqli->prepare("select username from user\_user where id=?"))

{

/\* ligar parámetros para marcadores \*/

$stmt->bind\_param("i", $\_SESSION['\_uid']);

/\* ejecutar la consulta \*/

$stmt->execute();

/\* ligar variables de resultado \*/

$stmt->bind\_result($carnet);

/\* obtener valor \*/

$stmt->fetch();

/\* cerrar sentencia \*/

$stmt->close();

}else{

echo "Falló laejecucion: (" . $mysqli->errno . ") " . $mysqli->error;

}

$mysqli->close();

//echo $carnet;

$mysqli = new mysqli("localhost", "root", "", "cpfecys");

if ($mysqli->connect\_errno) {

echo "Falló la conexión con MySQL: (" . $mysqli->connect\_err . ")"

. $mysqli->connect\_error;

}

$token = sha1($carnet . time());

/\* crear una sentencia preparada \*/

if ($stmt =

$mysqli->prepare('update auth\_user set uv\_token=? where username=?')){

/\* ligar parámetros para marcadores \*/

$stmt->bind\_param('ss', $token, $carnet);

// $stmt->bind\_param('s', $username);

/\* ejecutar la consulta \*/

if(!$stmt->execute()){

echo "Fail";

}

$stmt->close();

}else{

echo "Fallo la ejecucion " . $mysqli->errno . " error: " . $mysqli->error;

return;

}

$mysqli->close();

//echo $token;

$ret = array();

$ret['token'] = $token;

$ret['uid'] = $\_SESSION['\_uid'];

//$ret['carnet'] = $carnet;

echo json\_encode($ret);

?>

Fuente: elaboración propia.

* + 1. Instalando aplicaciones en web2py

1. FASE ENSEÑANZA APRENDIZAJE
   1. Roles y accessos
      1. Roles

Los roles contenidos en CPFECYS son los siguientes:

* Cantidad de estudiantes participando en la actividad
* Administrador
* Tutor Académico
* Supervisor
* DSI

Y sus funciones y acciones son como se describen a continuaciónÑ

* + - 1. Administrador

El administrador cuenta con todos los permisos de creación de contenido y publicación del mismo, puede también enviar notificaciones y recordatorios a los integrantes del resto de grupos disponibles, así mismo tiene permisos de calificación y aprobación de cualquier contenido generado por los Tutores académicos, entre el contenido que el administrador puede generar se encuentra:

* Restricciones de reportes a entregar
* Reportes a entregar pre programados para una fecha
* Entregables requeridos por área de DTT
* Proyectos que son excepciones de entregables
* Ingresar usuarios nuevos
* Asignar roles y proyectos a usuarios
* Validar, aprobar, modificar y remover asignaciones
  + - 1. Tutor Académico
      2. Supervisor
      3. DSI

conclusiones

Recomendaciones

bibliografía

1.