Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ingeniería en Ciencias y Sistemas

**CREACIÓN DE SISTEMA DE COORDINACIÓN DE PRÁCTICA FINAL PARA LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

**Boris Aramis Aguilar Rodríguez**

**Walter Omar Vides Rodríguez**

Asesorado por el Inga. Gladys Sucely Aceituno

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

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**CREACIÓN DE SISTEMA DE COORDINACIÓN DE PRÁCTICA FINAL PARA LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA

FACULTAD DE INGENIERÍA

POR

**BORIS ARAMIS AGUILAR RODRIGUEZ**

**WALTER OMAR VIDES RODRIGUEZ**

ASESORADO POR LA ING. GLADYS SUCELLY DE ACEITUNO

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:

**CREACIÓN DE SISTEMA DE COORDINACIÓN DE PRÁCTICA FINAL PARA LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

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 VI

LISTA DE SÍMBOLOS VII

GLOSARIO VIII

RESUMEN XI

OBJETIVOS XII

INTRODUCCIÓN XIII

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 7

2.2.3. Control de estudiantes 7

2.2.4. Control de proyectos 7

2.2.5. Integración con Chamilo 8

2.2.5.1. Rol de catedrático y rol de practicante final 8

2.2.5.2. Rol de administrador de sistema 9

2.2.5.3. Rol de DSI 9

2.2.6. Limitantes de la Integración con chamilo 9

2.2.7. Control y seguimiento de proyectos 10

2.2.7.1. Reporte de resultados de actividad de curso 10

2.2.7.2. Bitácora de actividades efectuadas durante el mes 11

2.2.7.3. 11

2.2.7.4. Tareas a entregar como archivo 11

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

2.2.7.6. Tareas que se califican con nota 12

3. FASE TÉCNICO PROFESIONAL 13

3.1. Implementación de Funcionalidad para Control de Práctica Final en Escuela de Ciencias y Sistemas 13

3.1.1. Administración de Áreas de DTT 13

3.1.1.1. Administración de proyectos 13

3.1.2. Administración de Usuarios 13

3.1.3. Activación de práctica final y asignación a proyecto 14

3.1.3.1. Carga de Practicante Final por medio de archivo 14

3.1.3.2. Primer Ingreso de practicante final 14

3.1.3.1. Entrega e Ingreso de Reportes 15

3.1.3.1.1. Reporte tipo Final 15

3.1.3.1.2. Reporte tipo Parcial 15

3.1.3.1.3. Segmentos de Reporte Final 16

3.1.3.2. Entrega e Ingreso de Requisitos 16

3.2. Configuración de servidor de producción 16

3.2.1. Instalación de Apache 17

3.2.2. Instalación de Python 18

3.2.2.1. Configuración de módulos de Apache para Python 2.7 19

3.2.3. Instalación de Web2py 19

3.2.3.1. Instalación de Web2py desde la fuente oficial 19

3.2.3.2. Configuración de Apache para servir la aplicación 20

3.2.3.3. Configuración de directorios de errores y accesos 21

3.2.4. Instalación de Base de Datos 21

3.2.5. Integración con Chamilo 22

3.2.6. Instalando aplicaciones en Web2py 26

3.3. Costos y Recursos 28

3.3.1. Costos 28

3.3.2. Recursos 29

4. FASE ENSEÑANZA APRENDIZAJE 29

4.1. Beneficios del proyecto CPFECYS 29

4.2. Capacitación 30

4.3. Diagrama de Roles 30

4.4. Roles y accesos 33

4.4.1. Roles 33

4.5. Acciones de Administrador 33

4.5.1.1. Administración de roles 34

4.5.1.2. Administrador 35

4.5.1.3. Tutor Académico 37

4.5.1.4. Supervisor 38

4.5.1.5. DSI 40

4.5.1.6. Creación de contenido por rol 42

4.5.1.7. Enlaces 43

4.5.1.8. Notificaciones 43

4.5.1.9. Envió de correos a usuarios por rol y proyecto 43

4.5.1.10. Carga de usuarios 44

4.5.1.11. Carga de usuarios por archivo 44

4.5.1.12. Asignación de usuarios 45

4.5.1.13. Planificación de reportes 45

4.5.1.14. Planificación de entregables 45

4.5.1.15. Aprobación de practicas 46

4.5.1.16. Reportes y estadísticas 46

4.5.1.17. Administración de DSI 46

4.5.1.18. Generar constancia final 47

4.5.1.19. Procesos automatizados 47

4.5.1.20. Creación de reportes de manera automática 47

4.6. Aprobación de practica 47

4.7. Finalización de ciclo 49

4.8. Reportes y estadísticos 49

4.9. Constancias y finalización de práctica 50

4.10. Actividades de Supervisor en CPFECYS 50

4.10.1.1. Calificación de reportes 50

4.10.1.1.1. Borrador 50

4.10.1.1.2. Pendiente de Calificación 50

4.10.1.1.3. Calificado 51

4.10.1.1.4. En revisión 51

4.10.1.2. Revisiones 51

4.11. Actividades de tutor académico en CPFECYS 51

4.11.1.1. Ingresar reportes 51

4.11.1.2. Ingresar entregables 52

4.12. Comunicación 52

4.12.1. Notificaciones 52

4.12.1.1. Notificaciones por correo 53

4.12.1.2. Noticias 53

4.12.1.3. Notificaciones de reportes 53

Conclusiones 55

Recomendaciones 56

Bibliografía 59

ÍNDICE DE ILUSTRACIONES

**FIGURAS**

Figura 1……………………………………………………………………Página 17

Figura 2……………………………………………………………………Página 18

Figura 3……………………………………………………………………Página 19

Figura 4……………………………………………………………………Página 20

Figura 5……………………………………………………………………Página 20

Figura 6……………………………………………………………………Página 21

Figura 7……………………………………………………………………Página 22

Figura 8……………………………………………………………………Página 23

Figura 9……………………………………………………………………Página 24

Figura 10……………………………………………………………………Página 25

Figura 11……………………………………………………………………Página 27

Figura 12……………………………………………………………………Página 27

Figura 13……………………………………………………………………Página 28

Figura 14……………………………………………………………………Página 31

Figura 15……………………………………………………………………Página 31

Figura 16……………………………………………………………………Página 34

Figura 17……………………………………………………………………Página 35

Figura 18……………………………………………………………………Página 37

Figura 19……………………………………………………………………Página 38

Figura 20……………………………………………………………………Página 39

Figura 21……………………………………………………………………Página 39

Figura 22……………………………………………………………………Página40

Figura 23……………………………………………………………………Página42

Figura 24……………………………………………………………………Página42

Figura 25……………………………………………………………………Página 45

**TABLAS**

Tabla 1……………………………………………………………………Página 28

LISTA DE SÍMBOLOS

**Símbolo Significado**

GLOSARIO

|  |  |
| --- | --- |
| **Web2py** | Web2py es un marco de trabajo o *Framework* basado para crear aplicaciones web, está basado en el lenguaje de programación *Python* y utiliza metodología de desarrollo ágil. |
| **Metodología de desarrollo de software** | Una metodología de desarrollo es el conjunto de prácticas y técnicas utilizadas en la gestión de un proyecto, comprende desde la documentación generada hasta los medios para publicación del producto a los usuarios finales, contemplando temas como sistemas de pruebas y estilos de programación. |
| **CPFECYS** | CPFECYS es el nombre del proyecto de reingeniería del proyecto para coordinación de auxiliatura de los estudiantes de ciencias y sistemas (CAECYS), y su significado es coordinación de práctica final de estudiantes de ciencias y sistemas. |
| **CAECYS** | CAECYS es el nombre del proyecto destinado a gestionar el control de estudiantes en su practica final y su significado es coordinación de auxiliatura de estudiantes de ciencias y sistemas, es el predecesor de el proyecto presentado en este documento, CPFECYS. |
| **Modelo** | En la configuración de arquitectura nombrada como MVC por Modelo Vista Controlador, es la entidad que representa la información una sección de la información a almacenar, representa una entidad del modelo de negocio y puede ser por ejemplo el usuario. |
| **Vista** | En la arquitectura Modelo Vista Controlador la vista es la parte encargada de desplegar la información al usuario, puede ser también visto de manera más general como una interfaz que habilita la interacción con un usuario u otro ente. |
| **Controlador** | En la arquitectura Modelo Vista Controlador el controlador es la parte del código que gestiona todas las validaciones y reglas de negocio, es la parte lógica del código que procesa las entradas y prepara y devuelve las respuestas a la capa de la vista, el controlador puede efectuar desde ninguna hasta muchas interacciones con la capa de modelo. |
| **Marco de trabajo** | Un marco de trabajo también conocido como *Framework* es un conjunto de librerías y configuraciones que proveen a los desarrolladores de software de un entorno y herramientas que facilitan, estandarizan el proceso de producción de software, agilizándolo y proveyendo una base sólida en aspectos de seguridad, escalabilidad y mantenimiento del software. |
| **DTT** | DTT es el programa gestionado en la Universidad de San Carlos de Guatemala permite a los estudiantes de Ingeniería efectuar sus prácticas finales en actividades de soporte dentro de entidades dentro de la universidad, por medio de la gestión de disponibilidad de plazas y recursos humanos |

RESUMEN

En el presente trabajo, se plasma el trabajo efectuado en el EPS durante el cual se llevó a cabo el proceso de reingeniería de la herramienta utilizada para el control, seguimiento y comunicación de los proyectos de práctica final de los estudiantes de Ciencias y Sistemas.

La necesidad principal que impulsó la ejecución de este proyecto radica en la necesidad de expandir la funcionalidad de la herramienta con la que contaba al momento de iniciado el EPS, nombrada CAECYS la herramienta anterior, contemplaba solamente un área del programa DTT que es el programa que contienen todos los tipos de proyectos en los que un estudiante de la escuela puede involucrarse para hacer su práctica final, esto provocó que el seguimiento de los proyectos del resto de áreas fuese muy débil y los procesos manuales.

La nueva herramienta soporta completamente y de manera flexible los nuevos requerimientos de los proyectos del programa DTT, automatizando e integrando todos los procesos y comunicación de los mismos en una única herramienta y eliminando la necesidad de tener tareas manuales para verificar la validez de las actividades de los estudiantes.

Adicionalmente se crearon módulos para cumplir con funcionalidad con la que la aplicación anterior no contaba, este nuevo sistema fue diseñado buscando la máxima flexibilidad en todas las áreas del programa DTT y en todas las actividades en la medida de lo posible, para garantizar un mayor tiempo de vida de la herramienta.

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, son 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.

* + - 2. 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 1

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 2.

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 3.

Los módulos configurados en los comandos de la figura 3 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 4.

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 5.

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 6.

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 7.

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 8

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 figura 9.

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

Para poder instalar o hacer copias de seguridad de aplicaciones gestionadas en Web2py primero debe accederse a la url donde se encuentre configurado o bien ejecutarlo por medio del comando mostrado en la figura 10.

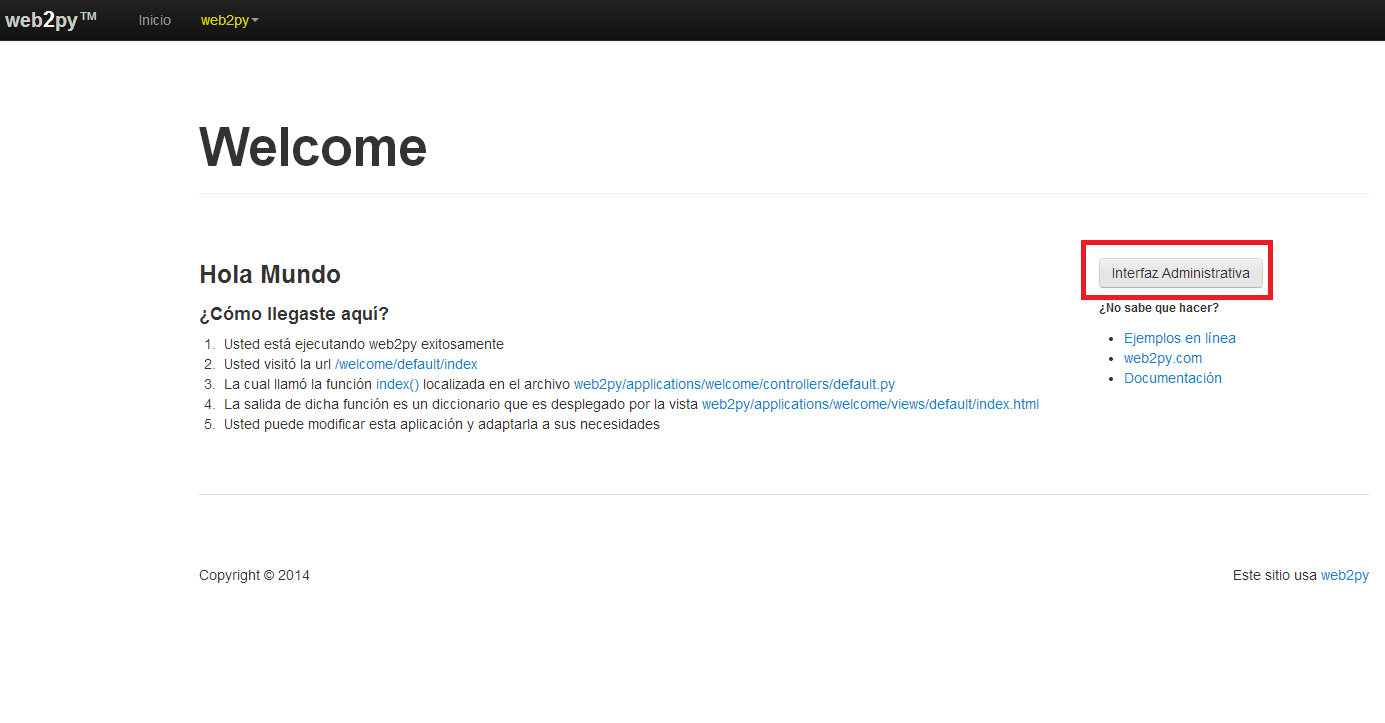
1. Contenido del archivo session\_handler

python Web2py.py

Fuente: elaboración propia.

Una vez Web2py se encuentre en ejecución mostrará una pantalla como la de la figura 11, donde se debe seleccionar la opción Interfaz Administrativa,

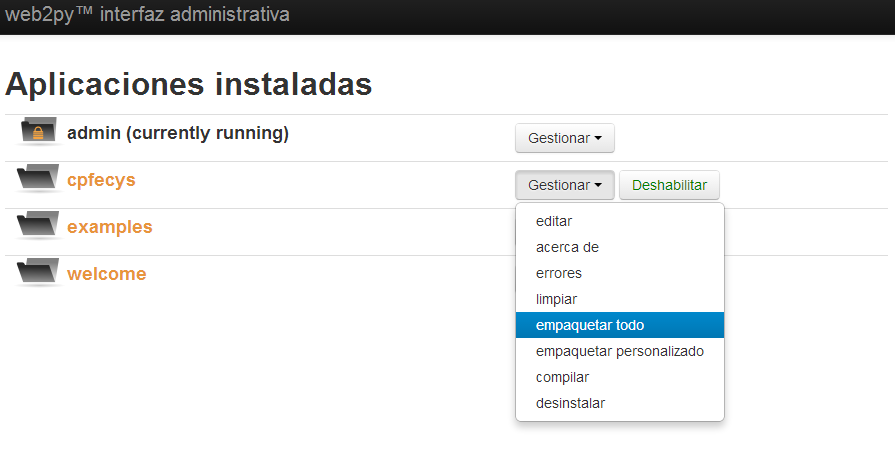
1. Web2py interfaz administrativa



Fuente: elaboración propia.

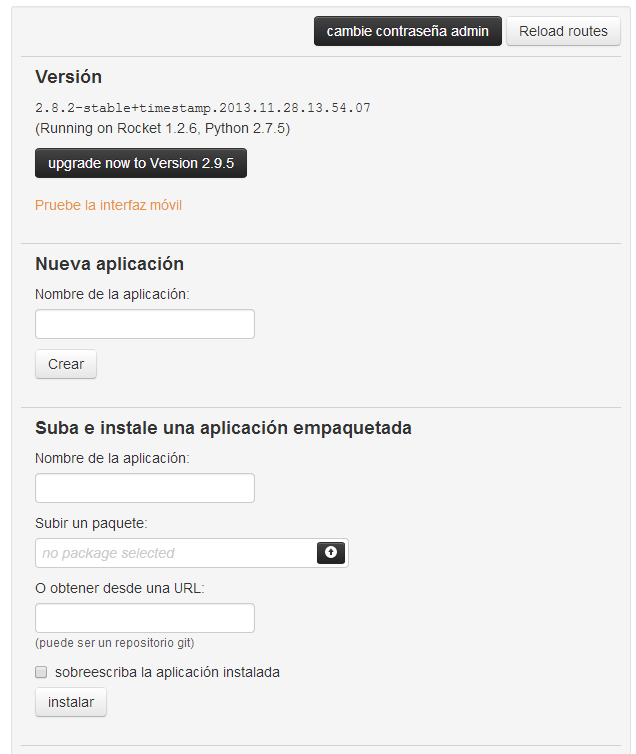
La interfaz administrativa contiene los controles para hacer un backup de la aplicación seleccionada, esta interfaz se muestra en la figura 12, también cuenta con la opción de cargar una aplicación empaquetada, esto se muestra en la figura 13.

1. Web2py hacer copia de una aplicación



Fuente: elaboración propia.

1. Web2py instalar una aplicación



Fuente: elaboración propia.

* 1. Costos y Recursos
     1. Costos

Los costos incurridos en el desarrollo del EPS consideran el tiempo de los desarrolladores implicados, costos de servidores para desarrollo, pruebas y producción y recursos consumidos durante el tiempo de ejecución del proyecto.

1. Tabla de Costos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Recurso** | **Cantidad** | **Costo (mes)** | **Tiempo (mes)** | **Total** |
| Desarrollador *Senior* | 2 | Q15,000 | 11 | Q 330,000 |
| Servidor cloud | 1 | Q80 | 11 | Q880 |

Fuente: elaboración propia.

* + 1. Recursos

Como recursos se consideran los recursos humanos y de hardware disponibles durante el proceso de realización del EPS, los recursos de los que se tuvo disponibilidad son

* Dos desarrolladores *Senior*
* Dos ordenadores portátiles, para el entorno de desarrollo
* Un servidor virtual *cloud* de renta mensual

1. FASE ENSEÑANZA APRENDIZAJE
   1. Beneficios del proyecto CPFECYS

Los beneficios de CPFECYS radican principalmente en las funciones adicionales a las que tiene la aplicación CAECYS, CAECYS es la aplicación predecesora a CPFECYS, las funciones nuevas disponibles son

* Soporte completo de la estructura DTT
* Flexibilidad para crecimiento del proyecto DTT
* Proceso de revisión de reportes entregados
* Posibilidad de expansión de la herramienta, integración e interacción con futuros proyectos
* Existencia de base de datos única y unificada y disponible para el uso de futuras aplicaciones de la escuela de ciencias y sistemas
* Facilidad de migración, configuración e instalación de la herramienta
* Módulos de notificaciones diseñados para filtrar por roles la información a enviar
* Portabilidad de Web2py
* Carga masiva de estudiantes y catedráticos
* Posibilidad de expandir la estructura DTT por medio de editar roles, áreas y proyectos
  1. Capacitación

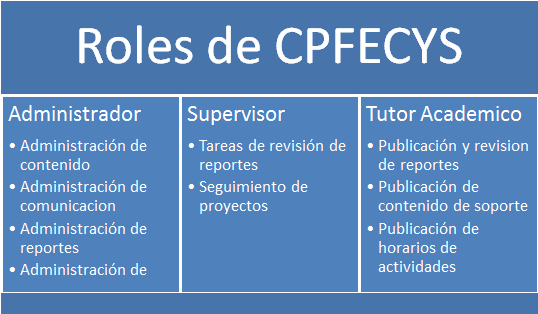
El proceso de capacitación durará un máximo de dos meses se dará a partir de la aprobación de finalización del proyecto y consistirá en soporte para el administrador de la herramienta y a los estudiantes encargados de dar continuidad al EPS, la manera de ejecutar dicha capacitación será por medio de reuniones presenciales con los involucrados, o bien por medio electrónico.

La capacitación de cara al resto de usuarios finales será por medio del manual de usuario creado y en caso de ser necesario y existir la posibilidad de tiempo por medio de video tutoriales explicativos que muestren los pasos para los procesos importantes de cada rol.

* 1. Diagrama de Roles

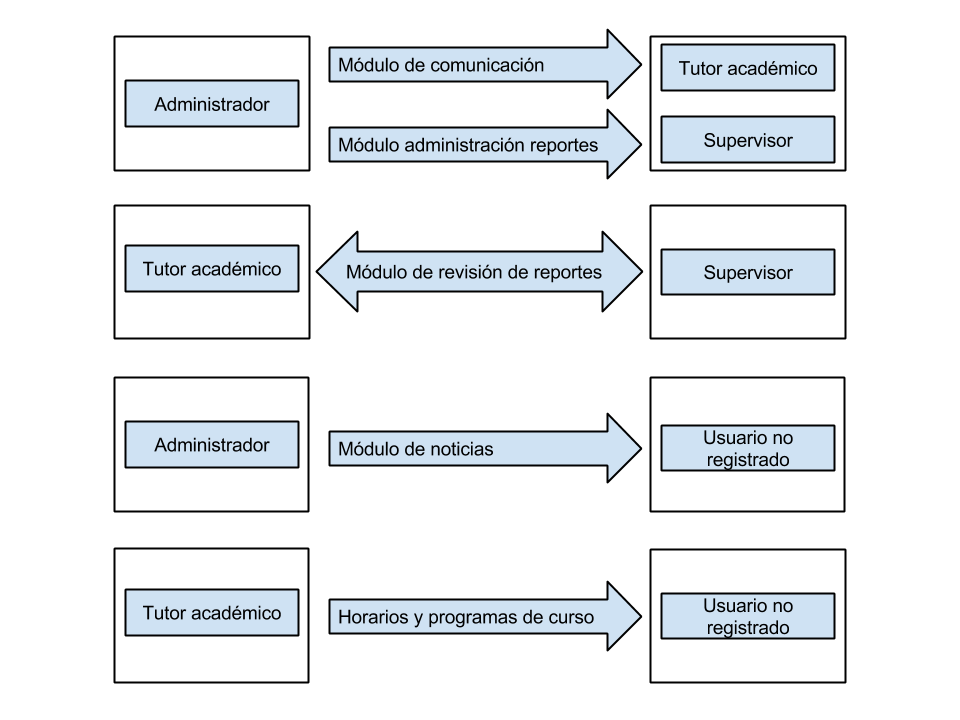
La distribución de tareas de un usuario de CPFECYS es dependiente del rol o roles que tenga asignados, las tareas operativas se distribuyen entre los roles supervisor y tutor académico, de la manera mostrada en la figura 14, en CPFECYS las interacciones entre los usuarios de diferentes roles suceden por medio de módulos o tareas disponibles a sus roles, en la figura 15, se muestran los medios de interacción entre los roles de CPFECYS

1. Diagrama de actividades de roles de CPFECYS



Fuente: elaboración propia.

1. Medios de interacción entre roles de CPFECYS



Fuente: elaboración propia.

La interacción en la dirección de administrador de la herramienta a tutor académico y supervisor se lleva a cabo principalmente por medio de dos módulos, el módulo de comunicación y el módulo de revisión de reportes, el módulo de comunicación es el que habilita al administrador para poder enviar notificaciones por correo electrónico a cualquier usuario delimitando por rol, la comunicación en este módulo es en una dirección, es decir de manera interna CPFECYS únicamente permite al administrador comunicar a los demás usuarios de la plataforma información de relevancia, este módulo incluye noticias, archivos, enlaces importantes y comunicación vía correo electrónico.

El otro medio de comunicación entre usuarios registrados es el que gestiona la interacción de supervisores con tutores académicos, la comunicación entre estos dos roles es bidireccional a diferencia de la comunicación entre administrador y resto de roles, aun cuando esta comunicación es bidireccional el contenido de los mensajes generados que van en la dirección de tutor académico a supervisor no pueden ser modificados o personalizados por el tutor académico, son mensajes automatizados que notifican al supervisor cuando una tarea ha sido completada, mientras que la comunicación en la dirección contraria, es decir de supervisor a tutor académico puede ser personalizada, en estos mensajes el supervisor indica al tutor académico requerimientos de cambios en sus reportes para que cumplan con las expectativas del supervisor.

Existen también los usuarios no registrados en la plataforma, son usuarios que pueden estar interesados en horarios de cursos, laboratorios, calificaciones y programas de cursos entre otros documentos relacionados con un curso en cuestión, los mensajes para estos usuarios son generados por los usuarios de dos roles en la plataforma, tutor académico y administrador, siendo el administrador el que cuenta con las herramientas para crear noticias publicas y el tutor académico quien genera contenido, horarios y documentos durante el progreso de su práctica, esta comunicación es unidireccional, pues los usuarios no registrados no pueden producir ni modificar contenido, únicamente consumirlo.

* 1. Roles y accesos
     1. Roles

La herramienta CPFECYS hace validaciones y restringe el acceso a únicamente los módulos que estén determinados para un rol en sí, múltiples roles pueden ser asignados a un usuario dándole de esta manera accesos a todos los módulos de todos los roles que desempeñe, de esta manera pueden existir en la plataforma múltiples administradores, así también puede existir un administrador que tenga rol de supervisor e inclusive rol de DSI o bien tutor académico, aun cuando en la lógica del negocio estas combinaciones de roles en un usuario no tienen sentido, en la práctica pueden ser de gran utilidad para permitir a un usuario analizar desde múltiples puntos de vista el estado y comportamientos de la aplicación, configuraciones como estas serán necesarias para desarrolladores, los roles contenidos en CPFECYS son los siguientes:

* Administrador
* Tutor Académico
* Supervisor
* DSI

Y sus funciones y acciones son como se describen a continuación

* 1. Acciones de Administrador

Todas las acciones de administrador presentadas requerirán que el usuario tenga accesos de administrador y que de manera previa inicie sesión como con dichos accesos en CPFECYS.

* + - 1. Administración de roles

CPFECYS permite al administrador tener completo control sobre los roles disponibles en la herramienta, permite crear, eliminar y modificar roles, para administrar los roles el usuario debe ser administrador en CPFECYS y vistar el enlace llamado Roles bajo el menú Admin de usuarios como se muestra, dicha pantalla y menú se muestran en la figura 16

1. Administrar los roles de CPFECYS



Fuente: elaboración propia.

La vista de administrador de roles cuenta con las opciones de Vista, Editar, Eliminar y Añadir, para añadir un nuevo Rol basta con pulsar el botón, la aplicación mostrará entonces una vista como la de la figura 17.

1. Administrar los roles de CPFECYS: Acción Añadir



Fuente: elaboración propia.

La vista de edición de roles cuenta con una alta similitud con la vista de creación, el único cambio perceptible es que, en los campos que aparecen vacíos en la creación, en el formulario de edición ya existe información previa, que es la que se desea editar.

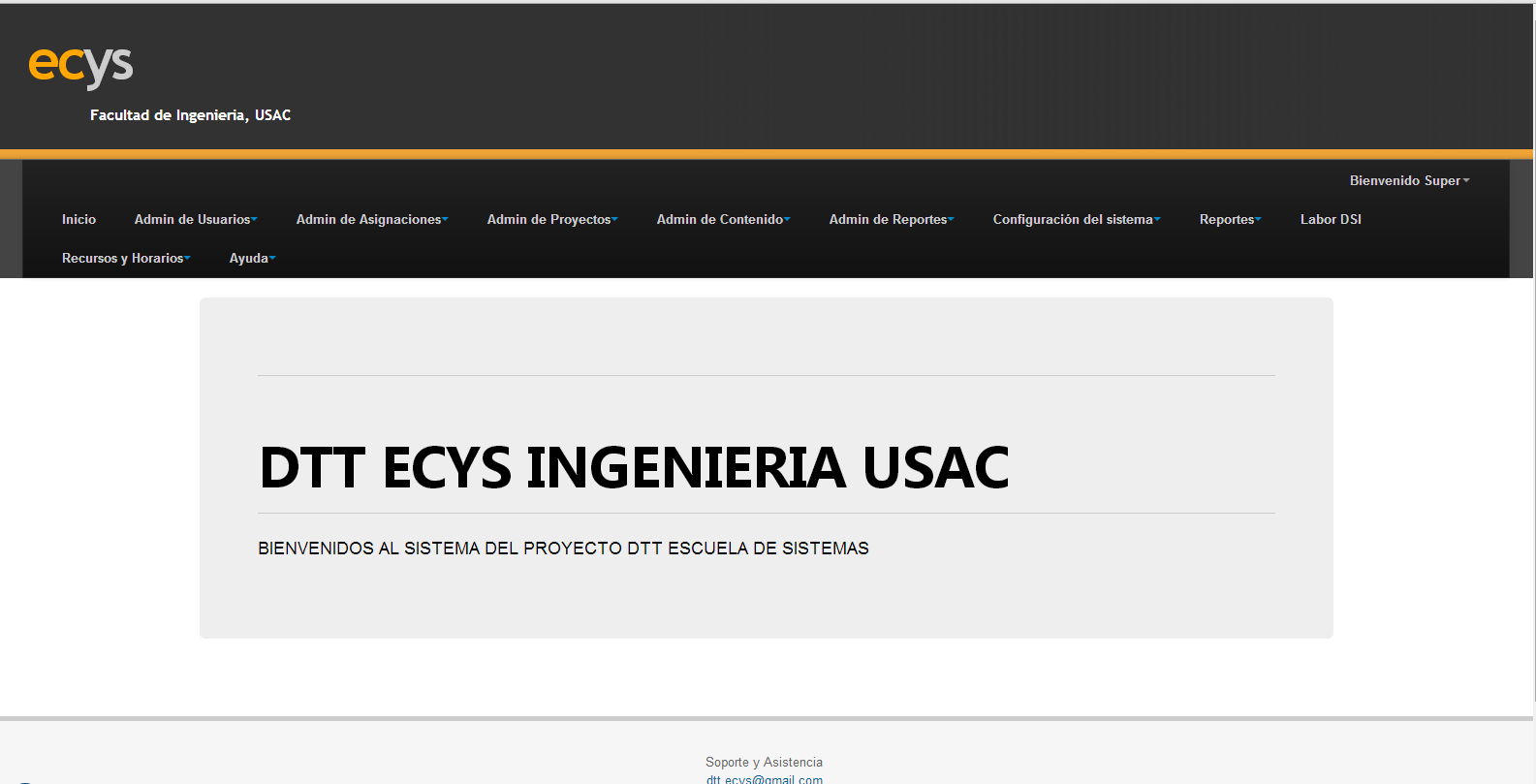
* + - 1. Administrador

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

La vista de la interfaz de administrador es la interfaz más extensa y con mayor cantidad de controles, menús y submenús, la configuración de la mismas se muestra en la figura 18.

1. Interfaz de Administrador



Fuente: elaboración propia.

* + - 1. Tutor Académico

Tutor académico es el rol que agrupa a los estudiantes que realizan su práctica final, el tutor académico tiene la responsabilidad de alimentar el sistema con estadísticos y resultados que justifiquen su práctica final y sirvan para análisis para los departamentos y entidades relacionadas con el proyecto DTT en la Universidad de San Carlos de Guatemala, las actividades que puede realizar un Tutor académico son las siguientes:

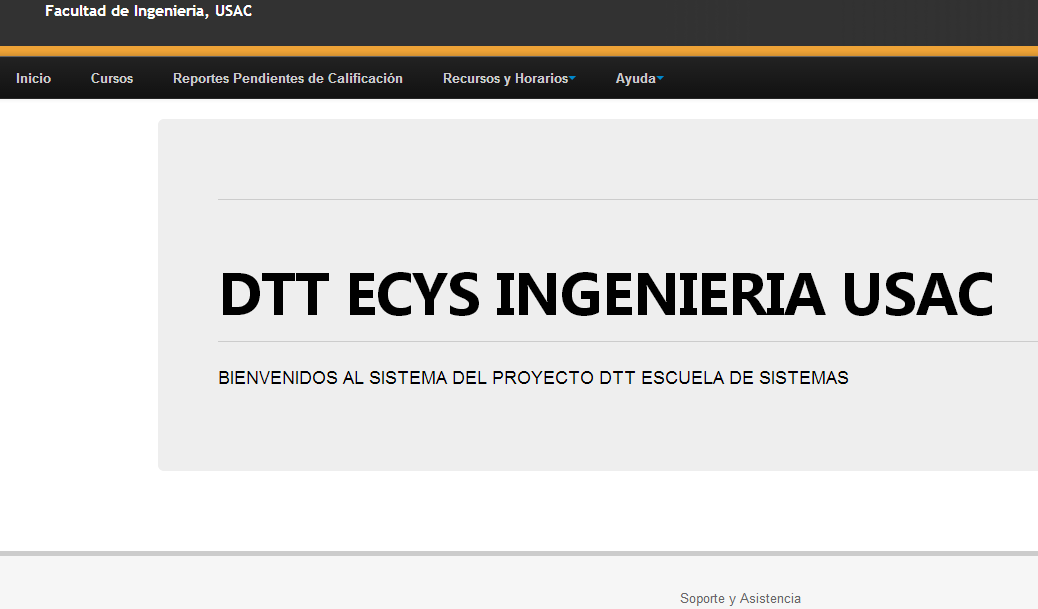
* Ingreso de reportes y estadísticas al sistema
* Ingreso de archivos y documentación solicitada para justificar actividades
* Entregables requeridos por área de DTT
* Generar reporte final
* Generar constancia de finalización
* Ingresar estadísticas de progreso de horas de practica
* Crear estadísticos de deserción de cursos
  + - 1. Supervisor

El rol de supervisor tiene como responsabilidad verificar y validar la información ingresada por los tutores académicos bajo su cargo, también tiene la responsabilidad de calificar la calidad de la información ingresada asignando una nota y un comentario a cada reporte ingresado por los tutores académicos bajo su cargo, las métricas disponibles para el supervisor son las siguientes

* Nota del reporte
* Comentario sobre el reporte visible para el estudiante

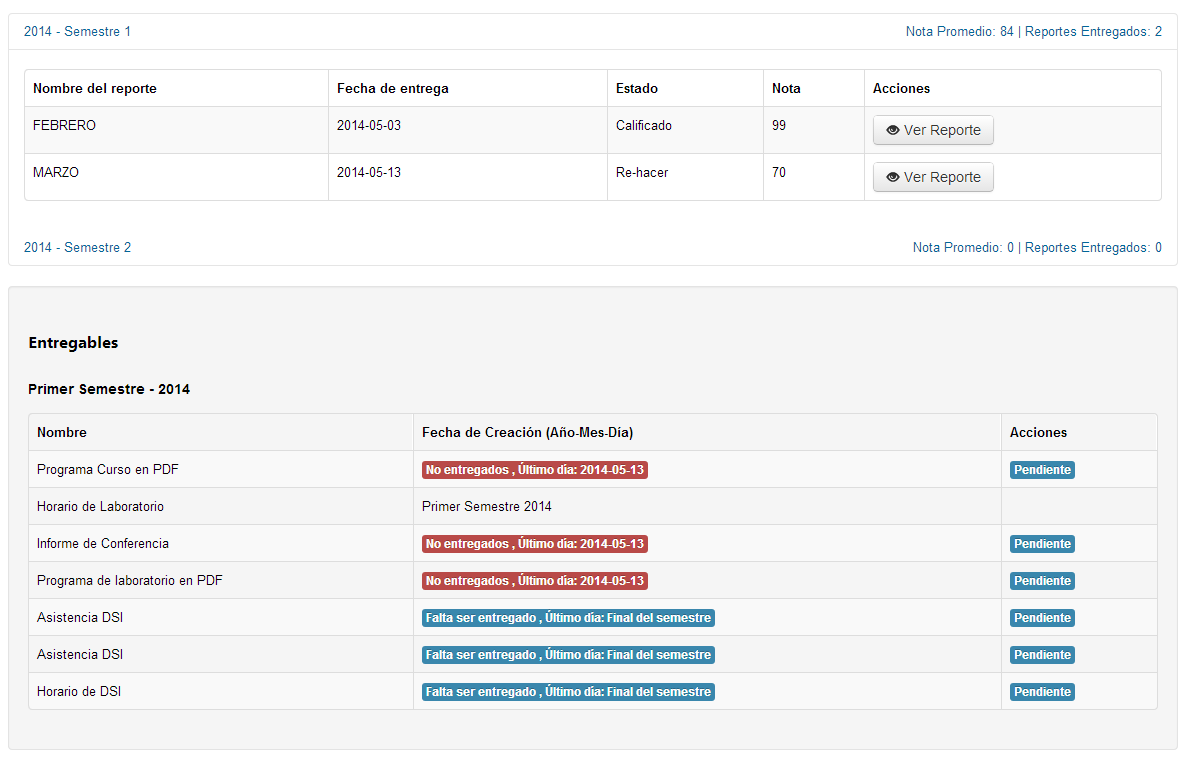
La interfaz de supervisor o jefe de proyecto cuenta con los controles y las vistas necesarios para manejar la comunicación y proceso de calificación de reportes enviados por los tutores académicos, estos controles se muestran a continuación en las figuras: figura 19, figura 20, figura 21.

1. Interfaz principal de supervisor



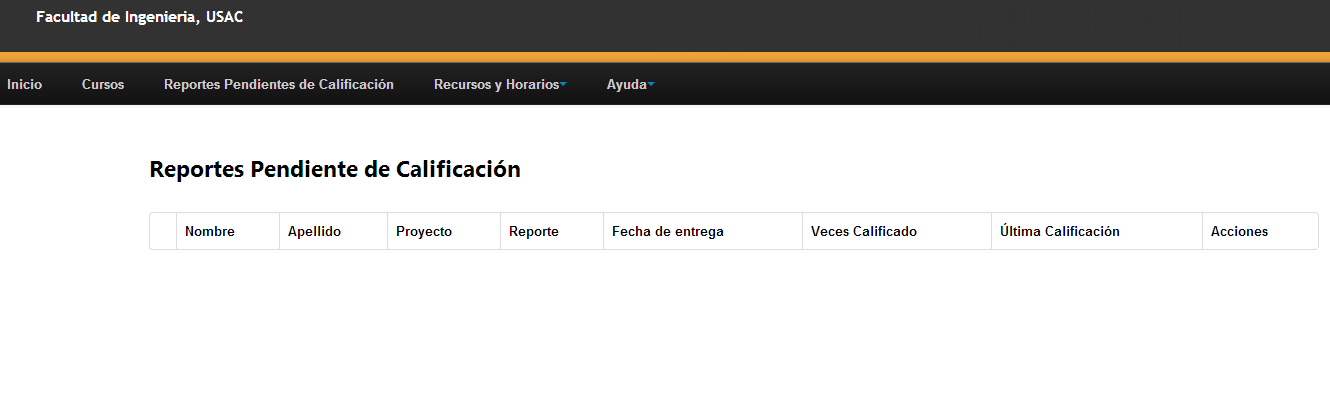
Fuente: elaboración propia.

1. Interfaz operativa, control de estudiante



Fuente: elaboración propia.

1. Interfaz operativa, reportes pendientes



Fuente: elaboración propia.

* + - 1. DSI

DSI es el rol destinado a dar soporte al administrador del sistema, el alcance de las tareas de los usuarios asignados a este rol son listadas a continuación:

* Verificación de actividades a realizar por un estudiante
* Calificación de asistencia en DSI
* Calificación de actividades asignadas a practicantes

Para acceder a los controles del rol DSI es requerido contar con las credenciales de un usuario que cuente con rol DSI asignado, una vez iniciada la sesión, el usuario contará con la opción Labor DSI como se muestra en la figura 22.

1. Interfaz DSI: Menú Labor DSI



Fuente: elaboración propia.

Luego de seleccionar el menú Los controles para las actividades se mostrarán al usuario, estos controles pueden ser modificados conforme funcionalidad sea agregada al rol DSI, la actividad principal actual del rol DSI es colocar nota o dar por aprobadas actividades de los tutores actividades que el Administrador decida delegar a los estudiantes con rol DSI asignado.

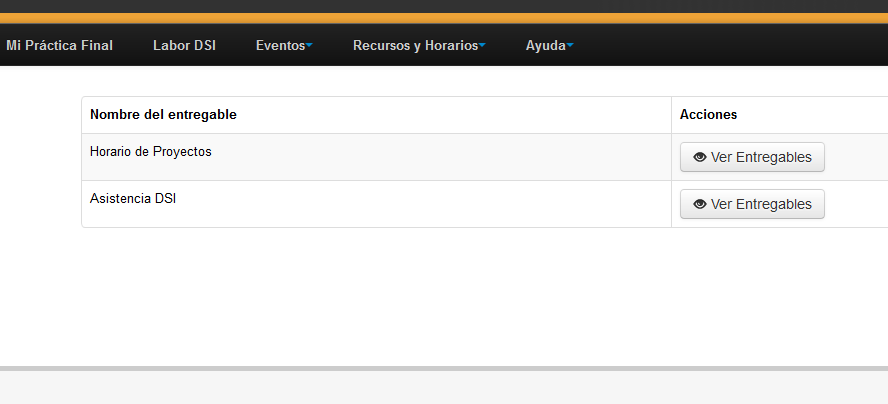
Los controles para aprobar una actividad delegada por el administrador a DSI dependen del tipo de la actividad en cuestión, las actividades delegadas pueden ser de dos tipos.

* Actividad con nota
* Actividad sin nota

Las actividades con nota cuentan con una nota mínima para poder ser consideradas como aprobadas, esta nota mínima es definida por el administrador y puede ser modificada en cualquier momento, la actividad sin nota solo se considera como realizada o no, donde realizada significa aprobada.

La lista de actividades disponibles en el semestre activo se despliega al usuario con rol administrador como se muestra en la figura 23.

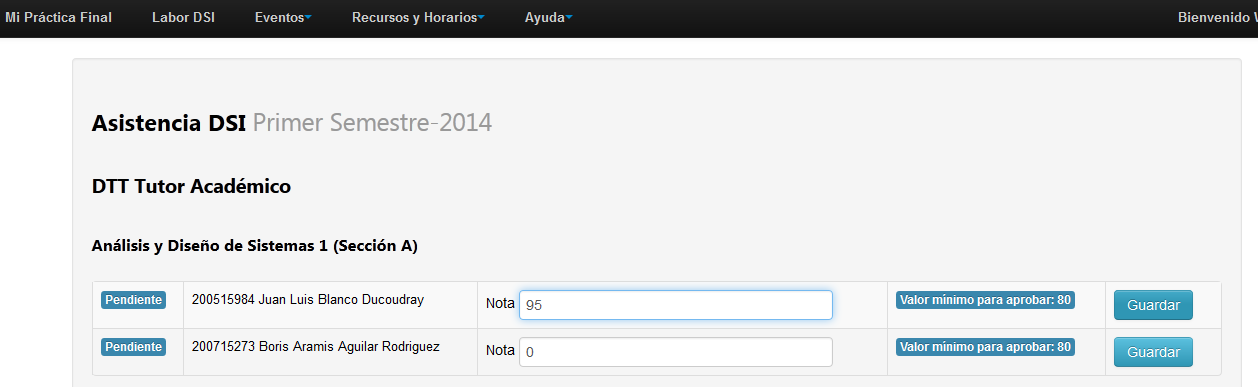
1. Interfaz DSI: Menú Labor DSI



Fuente: elaboración propia.

Una vez es seleccionada una de las actividades, la aplicación muestra al usuario el listado de todos los estudiantes que deben cumplir con dicha actividad en el semestre actual, los controles mostrados dependerán del tipo de la actividad, en la figura 24 se muestra el listado de una actividad de tipo actividad con nota.

1. Interfaz DSI: Menú Labor DSI

v

Fuente: elaboración propia.

* + - 1. Creación de contenido por rol

CPFECYS provee al administrador la herramienta para crear contenido visible a los usuarios asignándolo a un rol en específico, esto significa que según las reglas definidas por el administrador una publicación en específico podrá ser vista por usuarios que pertenecen por ejemplo, al rol tutor académico, mientras que para un usuario de rol Supervisor no será visible.

* + - 1. Enlaces

La categoría enlaces en el área de creación de contenido de CPFECYS permite al administrador publicar enlaces de interés para los usuarios y definir a que enlaces tiene acceso cada rol, el administrador puede definir un enlace en el menú administrador de contenido, seleccionando el submenú administración de enlaces, al seleccionar dicha interfaz se muestran al usuario los controles para crear, eliminar y editar un enlace, así como los roles para los cuales estará disponible dicho enlace.

* + - 1. Notificaciones

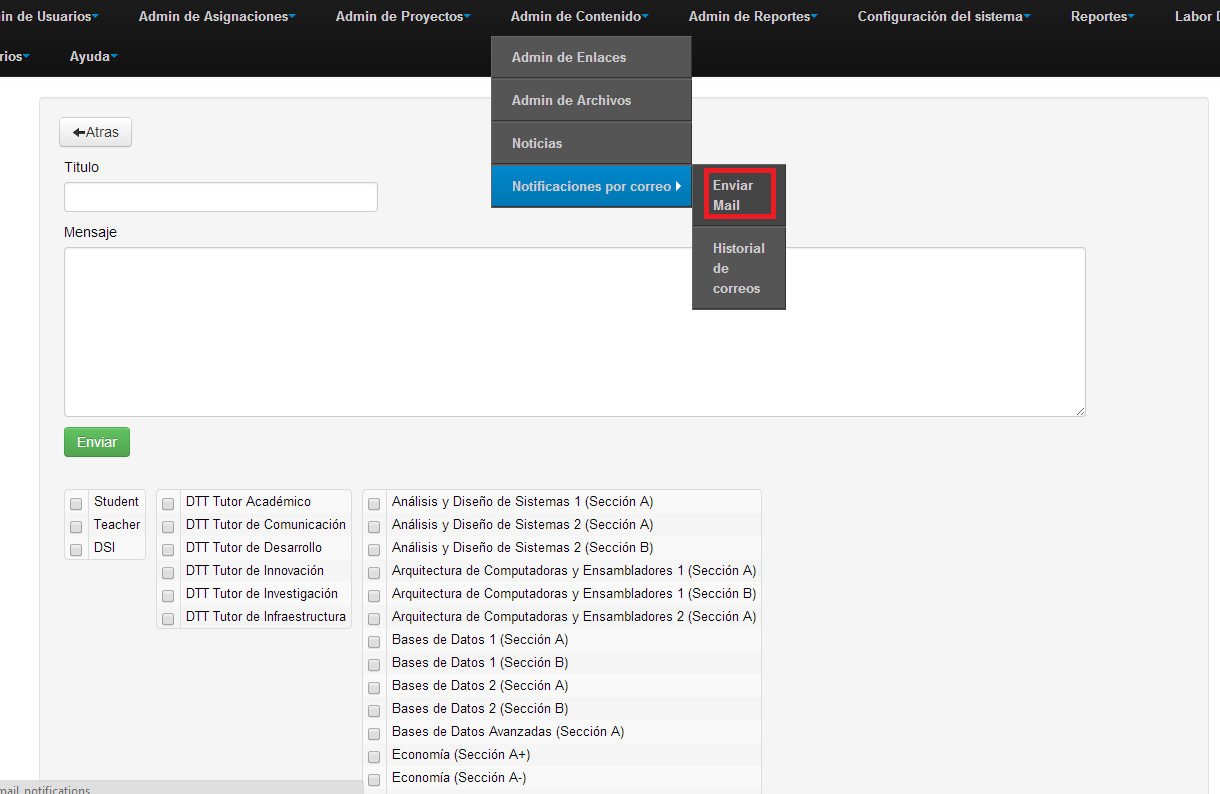
El usuario encargado del programa DTT está a cargo de gestionar la comunicación con los usuarios de la aplicación, por esto, la aplicación CPFECYS provee al administrador la posibilidad de publicar información importante en la página de inicio que es dependiente del rol del usuario que se encuentra en sesión.

* + - 1. Envió de correos a usuarios por rol y proyecto

Para gestionar la comunicación y notificaciones de eventos próximos relacionados con áreas o proyectos en específico CPFECYS cuenta con la herramienta de envío de correos para el administrador del sistema que permite notificar a usuarios registrados en la aplicación por medio de correos sobre actividades próximas y actividades en general.

La acción de enviar una notificación por correo a los usuarios de la plataforma se encuentra en el menú llamado Admin de contenido, en la sección Notificaciones por correo y la subsección enviar mail, en la figura 25 se muestran los controles para componer el mensaje a enviar en la parte superior y en la parte inferior se muestran los controles que permiten al administrador elegir quienes serán los receptores de dicho mensaje, una vez el mensaje esté listo para ser enviado y filtrado por roles.

1. Envió de correos a usuarios registrados



Fuente: elaboración propia.

* + - 1. Carga de usuarios

La carga de usuarios puede ser efectuada de dos maneras, una es por medio de la carga de un archivo csv que cuenta con información importante de los usuarios, la segunda forma es por medio de los controles contenidos en la interfaz de administrador.

* + - 1. Carga de usuarios por archivo

Los datos que deben contener los siguientes datos

* Nombre de usuario
* Apellidos del usuario
* Carnet del usuario
* Código del curso al cual será asignado
* Tiempo durante el cual estará
* Si la práctica será a ad honorem
  + - 1. Asignación de usuarios

La asignación de usuarios es automática cuando la carga es por medio de un archivo de carga de usuarios, de manera alternativa un usuario puede ser asignado a uno o más proyectos, dichas asignaciones tienen un tiempo de duración que debe ser determinada por el administrador.

* + - 1. Planificación de reportes

La planificación de los reportes consiste principalmente en pre programar las fechas áreas y excepciones para los reportes a entregar por los tutores académicos y que luego serán calificados por los supervisores, los reportes programados estarán habilitados durante el tiempo que sea indicado por el administrador, tiempo durante el cual los estudiantes podrán crear los reportes y enviarlos, luego de este punto el supervisor podrá calificarlos, aprobándolos, reprobándolos o bien enviándolos a revisión para mejoras

* + - 1. Planificación de entregables

La interfaz de planificación de entregables consiste en definir fechas para un entregable, que debe ser creado por los tutores académicos que se encuentran en proyectos de las áreas a las que está asignado un entregable, los entregables puede ser de varios tipos, entre los cuales se encuentran

* Entregables de tipo archivo
* Entregables de tipo actividad emergente
* Entregables de tipo actividad con nota
  + - 1. Aprobación de practicas

El proceso de aprobación de la práctica es calculado por CPFECYS de manera automática, para casos especiales la herramienta provee al administrador la opción de aprobar una práctica omitiendo los requerimientos mínimos.

* + - 1. Reportes y estadísticas

Los reportes y estadísticas se generan de manera automática, y son dependientes de la información que se genera conforme los usuarios utilizan el sistema, el listado de reportes se muestra a continuación:

* Reporte de catedráticos activos
* Reporte que condensa los informes por estado
* Reporte de tutores académicos activos
* Reporte de supervisores activos

Las estadísticas generadas son las siguientes:

* Estadísticas de deserción
* Estadísticas individuales por tipo de actividad
* Estadísticas de alumnos presentes por actividad
* Estadísticas de notas y distribuciones por actividad
  + - 1. Administración de DSI

Las actividades de administración de DSI consiste en asignación y des asignación de usuarios al rol de DSI, la herramienta también provee de controles para cumplir con las actividades de los usuarios del rol DSI sin profundizar en el proceso.

* + - 1. Generar constancia final

La constancia de finalización de práctica es calculada y validada de manera automática por la aplicación CPFECYS, para casos especiales el administrador cuenta con los controles para forzar la autorización de este proceso y generar la constancia final, el requerimiento para generar la constancia para el sistema es que el tutor académico en cuestión haya cumplido con todos los reportes y entregables requeridos.

* + - 1. Procesos automatizados

CPFECYS cuenta con varios proceso que necesitan automatización en uno o más pasos, el marco de trabajo Web2py utilizado para desarrollar la aplicación cuenta con tareas repetitivas pre programadas llamadas *Scheduleres* cuyo trabajo es ejecutar tareas determinadas que se cumplen cuando componentes del sistema llenan ciertos requisitos, las actividades programadas para realizarse de manera automática son las siguientes:

* Crear de manera automática los reportes que no fueron ingresados en el lapso de tiempo límite
* Finalizar el semestre cuando se alcanza la fecha máxima determinada por el administrador
  + - 1. Creación de reportes de manera automática

Cuando un tutor académico falta en el cumplimiento de un reporte en el tiempo límite, la aplicación de manera automática al finalizar el último día para crear el reporte lo crea con nota de cero puntos, reprobado y con una nota que indica el motivo por el cual fue reprobado.

* 1. Aprobación de practica

El proceso de aprobación de práctica está basado en los requerimientos creados por el administrador a lo largo de un semestre, esto incluye, los reportes programados por el administrador para el semestre en cuestión, los entregables programados dentro de los semestres que tenga validez una práctica asignada.

El proceso de verificación de estos requerimientos es automático y funciona para todos los usuarios activos en la herramienta que cuenten con rol tutor académico, la sección de código encargada de esta actividad se encuentra dentro del programador de tareas automatizadas con el que cuenta Web2py.

Para soportar casos extraordinarios una vez iniciada la sesión de usuario administrador, bajo el menú nombrado Admin de asignaciones, se encuentra la opción nombrada asignaciones por semestre, donde el administrador puede tomar decisiones sobre cada asignación de manera individual, las acciones que puede tomar sobre las asignaciones son las siguientes

* Desactivar una asignación
* Activar una asignación
* Ignorar una asignación
* Marcar como fallida una asignación
* Marcar como exitosa una asignación

Una asignación puede ser marcada como ignorada en los casos en que por algún motivo se anule un semestre o práctica de un tutor académico y se decida extender el periodo de asignación del mismo para evitar la anulación total de su práctica.

Los casos de aprobación o reprobación de una asignación son reservados especialmente para casos en que por algún motivo la práctica debe ser reprobada o aprobada por el administrador, generalmente la aprobación manual se requiere cuando hay un precedente de conflicto con el supervisor del proyecto en el que el tutor académico se vea afectado de manera injustificada, mientras que la reprobación de una práctica queda a criterio del administrador.

* 1. Finalización de ciclo

La finalización de ciclo funciona de manera automática, y funciona por medio de un parámetro definido por el administrador, en el cual define la fecha de finalización, el administrador también puede terminar de manera manual el ciclo.

Una vez un ciclo es finalizado manual o automáticamente todas las asignaciones activas al momento que el ciclo finaliza, son congeladas y sus componentes no pueden ser editados por los tutores académicos, este es el primer paso antes de la finalización y validación de las asignaciones, una vez las asignaciones han sido congeladas CPFECYS puede hacer las validaciones correspondientes que generan los estados de aprobación y reprobación de las practicas activas.

* 1. Reportes y estadísticos

CPFECYS cuenta con una extensa cantidad de reportes y estadísticos de utilidad para el administrador, los reportes mínimos existentes son

* Listado de reportes generados, clasificados por estado
* Anomalías por periodo
* Descargar los entregables
* Reporte de cursos
* Reporte de supervisores de proyectos
* Reporte gerencial
* Registros de emails de todo el sistema

Estos reportes cuentan en algunos casos con estadísticas de comportamiento, toda la información es generada haciendo uso de la información de los informes ingresados por los tutores académicos, también de la información de los usuarios asignados en la plataforma y filtrados por su rol en la misma.

* 1. Constancias y finalización de práctica

Una vez el proceso de congelamiento y validación de prácticas es ejecutado por CPFECYS los usuarios con rol académico que cumplieron con todos los requerimientos a entregar durante la cantidad de ciclos asignados a su práctica contaran con la opción de generar la constancia de finalización de práctica, que servirá para iniciar el proceso de finalización, gestionado por el director del programa DTT, esta constancia tiene el detalle de la información generada por el estudiante y las notas de sus informes, información personal del alumno, información del curos, tiempo y ciclos en que ejecuto su práctica e información del supervisor a cargo.

* 1. Actividades de Supervisor en CPFECYS
     + 1. Calificación de reportes

El proceso de creación y calificación de reportes cuenta con cuatro posibles estados, que son, borrador, pendiente de calificación, calificado, en revisión.

* + - * 1. Borrador

Un reporte se encuentra en estado borrador cuando el tutor académico está trabajando en él, en este estado puede ser visto por el supervisor pero no puede ser aun calificado.

* + - * 1. Pendiente de Calificación

Cuando el tutor académico da por finalizado el reporte lo envía a calificación, este es el reporte pendiente de calificación y permanece en este estado durante todo el semestre hasta el momento en que el catedrático

* + - * 1. Calificado

Una vez el supervisor asigna una nota final al reporte y decide que la nota asignada es la final, a partir de este punto el reporte no puede ser editado ni calificado a menos que el administrador lo habilite para el caso necesario, adicionalmente es requerida la aprobación del director de DTT para que el reporte pueda ser considerado como válido.

* + - * 1. En revisión

Cuando el supervisor decide que el reporte no cumple con sus expectativas puede decidir retornar el reporte para que sea revisado por el tutor académico que lo creó, cuando la nota para revisión indique que el reporte está reprobando será obligatorio para el supervisor indicar al tutor académico el motivo por el cual el reporte fue rechazado.

* + - 1. Revisiones

El proceso de revisión consiste en dar la oportunidad al tutor académico de hacer mejoras en su reporte, esto implica que la calidad de un reporte y la información que contiene puedan mejorarse, también da un poder de arbitrariedad más amplio al jefe de proyecto y mejora la sinergia entre ambos roles por medio de la comunicación necesaria para dicho proceso.

La cantidad de revisiones posibles parametrizable y definida por el administrador de la herramienta, y puede ser modificada en cualquier momento, esta regla también puede ser reescrita e ignorada por el administrador cuantas veces el necesite.

* 1. Actividades de tutor académico en CPFECYS
     + 1. Ingresar reportes

El ingreso de reportes de tutores académicos es la fuente principal de alimentación de información estadística para la escuela de Ciencias y Sistemas de los proceso de práctica final y rendimiento de estudiantes en los cursos, tendencias y puntos de interés.

La tarea de los tutores académicos de ingreso de reportes consiste en registrar los valores estadísticos de las actividades planificadas y ejecutadas por ellos y sus jefes de proyecto durante el semestre y a lo largo del ciclo en que estén asignados para trabajar juntos en un curso o práctica.

Las secciones de información requeridas por los reportes son dependientes del área de DTT a la que pertenece dicho proyecto, la naturaleza de las actividades del área del proyecto en cuestión es el determinante del tipo de actividades que pueden ser medias.

* + - 1. Ingresar entregables

Los entregables son de utilidad tanto para estudiantes como para jefes de proyecto y usuarios administrativos, son también útiles para toma de decisiones y reportes de contenido, son utilizados para analizar y dar seguimiento a las actividades efectuadas en los cursos.

Estos entregables son generados por los tutores académicos y son entregados en formato pdf o doc en la plataforma CPFECYS, cuentan con un lapso de tiempo de entrega máximo que abarca el total del tiempo del semestre.

Los entregables de tipo público pueden ser descargados por cualquier usuario que tenga acceso a la plataforma, al igual que los de tipo horario que son visibles a cualquier usuario, el fin de esta configuración de privacidad es que estudiantes asignados o interesados en el curso puedan avocarse a la plataforma para consultar horarios o programas de curso.

* 1. Comunicación
     1. Notificaciones

CPFECYS cuenta con tres principales flujos de comunicación entre usuarios de la plataforma, el flujo principal es del administrador a los usuarios registrados en la plataforma categorizados por roles, también existe el área de noticias y el modulo dedicado a gestionar la comunicación entre jefes de proyecto y tutores académicos.

* + - 1. Notificaciones por correo

Las notificaciones por correo son de una via, y son enviadas por el administrador, donde él puede filtrar por área, por proyecto y por rol a los usuarios registrados en la plataforma, el administrador puede definir un asunto para el correo a enviar y el cuerpo del correo mismo.

Éste módulo de comunicación permite al administrador notificar cuando un reporte esté listo para su entrega, o bien un entregable para su ingreso, permite enviar recordatorios de tiempo próximo a expirar de un reporte o un entregable.

Cualquier comunicación de relevancia que sea necesario hacer llegar a los usuarios puede ser filtrada de manera predefinida y enviada haciendo uso de este módulo.

* + - 1. Noticias

El fin de este módulo es publicar en la página de inicio de CPFECYS donde basado en los roles del estudiante se mostrara contenido de interez para sus actividades.

El filtro del contenido es definido por el administrador en el momento en que crea una nueva publicación, este módulo se comporta por lo tanto como un gestor de contenido

* + - 1. Notificaciones de reportes

Las notificaciones de reportes son el flujo de información que fluye entre jefe de proyecto y tutor académico, este flujo se da principalmente cuando existe retroalimentación del tutor académico hacia el estudiante de la calificación y puntos a mejorar de los reportes entregados.

Dentro de este flujo también se encuentran los correos que se envían de manera automática en el momento en que el estudiante da por terminado un reporte y lo considera listo para ser entregado, al igual que la notificación por correo que se genera en el momento que el catedrático asigna una nota a un reporte y lo envía al estudiante con nota y observaciones.

Conclusiones

1. Aun cuando una aplicación es altamente adaptable, cuando hay cambios el modelo del negocio para el que fue diseñada y creada la aplicación, existen cambios necesarios en la aplicación, en estos casos cuando ya no se cuenta con el equipo que desarrollo la aplicación el proceso suele ser el de reingeniería este es el camino seguido con este proyecto para el programa DTT, el modelo de negocio y las operaciones fueron ampliadas y como consecuencia el requerimiento de una nueva aplicación fe creado.
2. La estandarización de los proceso y la expansión de las operaciones en el proyecto DTT excedió el sistema diseñado inicialmente para apoyar al administrador de la herramienta, esta es la principal justificación re hacer una herramienta de software.
3. Este proceso de reingeniería se dividió principalmente en dos fases, que fueron, primero, el proceso de depuración de la funcionalidad de la herramienta que se someterá a reingeniería y crear nuevamente la funcionalidad básica que persistirá con las mejoras solicitadas, luego la segunda fase, consiste en diseñar e implementar los nuevos requerimientos.
4. El entorno disponible para la implantación de un sistema puede ser un determinante importante del proceso de desarrollo en una solución de software, puede determinar la viabilidad de un proyecto y configuraciones únicas para un proyecto en cuestión y también pasos adicionales al proceso considerado desde el punto de vista del diseño.

Recomendaciones

1. Hacer uso de la versión de Centos 6.0 o superior para evitar tratar con problemas de compatibilidad con la librería Python, que de forma nativa en el Centos inferior a 6.0 requiere un proceso de actualización para permitir que Web2py funcione de manera correcta en el sistema operativo.
2. Crear un configurar un sistema de automatización de copias de seguridad que se ejecute de manera diaria, puede ser creado utilizando los *cron Jobs* del sistema operativo UNIX y haciendo uso también del comando mysqldump para volcar toda la información de base de datos a un archivo, luego via *smpt* hacer el envio de los archivos al mail del administrador de la plataforma.
3. No hacer uso de *phpmyadmin* bajo ninguna circunstancia, las operaciones sobre base de datos pueden ser efectuadas por medio de la terminal del sistema operativo y deben ser efectuadas de esa manera.
4. Cualquier proceso de migración de la aplicación debe ser efectuada copiando primero la totalidad de los archivos subidos por los usuarios y luego haciendo una copia de seguridad de la base de datos, implantando primero la base de datos en el nuevo ambiente seguido de los archivos de configuración.
5. La herramienta Web2py permite hacer la migración de la totalidad de un proyecto en un solo archivo empaquetado, esta es la mejor manera de migrar la aplicación y debe ser utilizada todas las veces que se necesite, migrar la información o bien replicar el ambiente de producción.
6. En el caso en que la herramienta de migración de Web2py por algún motivo no funcione correctamente o no permita migrar pueden ser copiados todos los archivos del directorio donde ha sido instalada la aplicación y seguir el manual de instalación para configurar el entorno necesario para el funcionamiento de la misma en el servidor en el que se desea efectuar la migración.
7. Hacer uso de un certificado SSL de encriptación de la información transmitida del cliente al servidor y viceversa para evitar ataques o robo de información de los usuarios registrados en la aplicación.
8. Utilizar un sistema de *scripts* que detecte el ingreso de administrador al servidor y notifique ya sea por email o mensaje SMS al teléfono o cuentas de correo electrónico de los administradores o administrador de la aplicación del ingreso para detectar posibles ingresos no permitidos, de esta forma se podrá saber de ataques o accesos no permitidos si el administrador recibe un mensaje de ingreso cuando no se ha programado alguno o se tiene el conocimiento que nadie debería estar en el servidor en ese momento.
9. En el caso en que sea necesario gestionar más aplicaciones en el servidor además de CPFECYS y que no tengan una relación directa con la aplicación, crear usuarios que cuenten únicamente con los permisos que correspondan a las necesidades del proyecto que requiera dicha implantación y permisos que aíslen al usuario de las áreas del sistema de archivos y base de datos relacionados con CPFECYS.
10. Habilitar los usuarios adicionales únicamente durante el tiempo prudente requerido y necesario para hacer las pruebas con las aplicaciones que requieran un espacio en el servidor donde se aloje CPFECYS y deshabiltiar los usuarios y sus permisos una vez sus tareas se cumplan.
11. Manejar de manera estricta las reglas de lo que un usuario podrá y no instalar, así como las reglas de que usuarios podrán instalar paquetes en el servidor.

Bibliografía

1. Python 2.7**.** [En línea] 3 de Julio de 2011. [Citado el: 7 de Octubre de 2013.] <2. https://www.python.org/download/releases/2.7 >.

2. *Installing Python Modules***.** [En línea] 15 de Enero de 2012. [Citado el: 7 de Octubre de 2013.] < https://docs.python.org/2.7/install/index.html >.

3. *Modwsgi.* [En línea] 15 de Marzo de 2011. [Citado el: 3 de Octubre de 2012.] <1. https://code.google.com/p/modwsgi/ >.

4. *Web2py workbook*. [En línea] 15 de Febrero de 2007. [Citado el: 10 de Octubre de 2013.] <http://www.web2py.com/book>.

5. *Introduction to Chamilo 2 structure*. [En línea] 2011. [Citado el: 15 de Agosto de 2013.] <https://support.chamilo.org/projects/chamilo-20/wiki/Introduction\_to\_Chamilo\_2\_structure >.

6. *How to report bugs*. [En línea] 2011. [Citado el: 15 de Agosto de 2013.] <https://support.chamilo.org/projects/chamilo-20/wiki/ReportingBugs>.

7. *Chamilo Communication Guide*. [En línea] 2005. [Citado el: 15 de Agosto de 2013.] <https://support.chamilo.org/projects/chamilo-20/wiki/ChamiloCommunicationGuide>.

8. *Highcharts General Documentation*. [En línea] 2014. [Citado el: 20 de Enero de 2014.] <http://www.highcharts.com/docs>.