



Facultad 4

Título: Desarrollo de la funcionalidad “Recolector” para el módulo de administración de Moodle 2.3.x

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

Autora: Liuvis Moreno Diaz

Tutores: Ing. Carlos Acosta Montejo

Ing. Suset Fernández Rojas

La Habana. Junio 2013

Pensamiento



“El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando; lo que más estamos sembrando son oportunidades a la inteligencia (...).” (Ruz, 1960)

Fidel Castro

Declaración de Autoría

Declaro ser autora del presente trabajo de diploma “Desarrollo de la funcionalidad “Recolector” para el módulo de administración de *Moodle 2.3.x*” y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmo la presente a los 25 días del mes de junio del año 2013.

Liuvis Moreno Diaz

Carlos Acosta Montejo

Suset Fernández Rojas

Datos de Contacto

Carlos Acosta Montejo: Ingeniero en Ciencias Informáticas, profesor Instructor. Se gradúa en el 2007 en la Universidad de las Ciencias Informáticas y comienza a trabajar en la dirección de Teleformación, en el 2009 comienza a laborar en el Centro FORTES, en ambos lugares se ha desempeñado como especialista de implantación administración y soporte de la plataforma Moodle, por lo que acumula 5 años de experiencia en el trabajo con dicha herramienta.

Suset Fernández Rojas: Ingeniera en Ciencias Informáticas, profesor Asistente. Se gradúa en el 2008 en la Universidad de las Ciencias Informáticas fecha desde la cual comienza a desempeñarse como especialista de la Dirección Técnica de la Infraestructura productiva en temas relacionados con el desarrollo colaborativo de software, además de impartir clases de Metodología de la Investigación Científica por 3 años. Actualmente se desempeña como especialista del Centro de Soporte UCI.

Agradecimientos

A mi mamá por nunca rendirse conmigo, por ser esa mujer fuerte de la cual vivo orgullosa, por guiarme siempre por el buen camino y sobre todo por ser mi compañera y amiga incondicional.

A mi papá gracias por confiar siempre en mí, por incitarme a continuar mis estudios, a ti te debo el logro de este sueño, ser una Ingeniera en Ciencias Informática, por complacerme en todos mis gustos y caprichos según han estado a tu alcance, te amo.

A todos mis hermanos Xiuvis, Rodolfo, Rando, Michel y Fulier, espero que se sientan orgullosos de mí.

A mis tíos Natalia y Treno por acogerme como una hija más y brindarme su cariño.

A mis mejores amigas Arianna y Suraima por contar con su bella amistad, porque más que amigas son mis hermanas y porque les prometí que me graduaría en nombre de las tres.

A mi amigo Marlon por tenerme paciencia y saber comprenderme en todo momento, por ser esa persona en la cual podía confiar siempre y con la que compartí hermosos momentos de mi vida que nunca olvidaré.

A mi amiga Durania por darme su apoyo y estar conmigo en los momentos buenos y en los no tan buenos también.

A Hilda y Osvaldo gracias por toda su ayuda y su preocupación por mí.

A mis amigos del grupo por saber ser mi familia cuando los necesite en estos 5 años, especialmente a Dianne, Daimaris y Augusto, gracias por su apoyo.

A mi amigo Denis por estar siempre pendiente de mis estudios y ser como un hermano más.

A los profesores que ayudaron en mi formación profesional, en especial Ulises.

A mis tutores Carlos y Suset por todo el apoyo ofrecido para la realización de este sueño, por estar siempre que necesite su apoyo y por convertirme en una mejor profesional.

A Carlos Alberto Hues porque sin ti no hubiera podido alcanzar esta meta, gracias por tu tiempo y dedicación.

A la revolución por darme la oportunidad de estudiar en este centro de excelencia.

Y a todas esas personas que confiaron siempre en mí y de una forma u otra ayudaron a mi formación y fueron parte de este logro en especial a toda mi familia.

Dedicatoria

A mis padres por ser las personas más importantes en mi vida, por darme su apoyo incondicional en los momentos que más necesitaba, por dedicar sus vidas a formarme como una mejor persona y guiarme por el buen camino.

A mis hermanos a los cuales amo con todo mi corazón y son mi razón de ser. A mi hermanito Rodolfo, espero que sigas mi ejemplo o seas mucho mejor.

A mi sobrinito Brayán por ser el niño de mis ojos, te quiero mucho.

A mi mejor amiga Arianna por creer siempre en mí y apoyarme en todas mis decisiones, aunque estés lejos de mí espero que siempre recuerdes que te quiero un montón.

Resumen

En la Universidad de las Ciencias Informáticas desde el año 2005 se utiliza la plataforma de teleformación *Moodle* como Entorno Virtual de Aprendizaje (EVA), un espacio para la enseñanza en línea y un importante apoyo al proceso de enseñanza aprendizaje. Actualmente los administradores del EVA en la Universidad exponen que algunos usuarios, recursos y actividades de los cursos existentes en el entorno perduran después de un largo tiempo, a pesar de no ser utilizados por sus usuarios. Además para consultar el estado de actividad de los usuarios, actividades y recursos, se debe seguir un proceso muy engorroso. El presente trabajo describe la implementación de la funcionalidad “Recolector” para el módulo de administración de *Moodle* en su versión 2.3.x, funcionalidad diseñada esencialmente con el objetivo de solucionar las dificultades antes mencionadas. El “Recolector” permitirá a los administradores del EVA listar los usuarios inactivos, enviarles una notificación de correo y eliminarlos completamente de la plataforma. Además mostrará las actividades o recursos pasivos de cada curso, facilitará el envío de notificaciones de correo a los profesores editores de los mismos y permitirá eliminar las actividades y recursos no utilizados luego de la confirmación necesaria. En esencia esta funcionalidad facilitará la toma de decisiones por parte de los administradores de la plataforma y posibilitará la liberación de espacio en la base de datos y servidores donde se encuentra la aplicación. El desarrollo estuvo guiado por la metodología *Programación Extrema (XP)* asistida por la herramienta Ingeniería de Software Asistida por Computadora (*CASE Visual Paradigm* 8.0). Se utilizó PHP como lenguaje de programación y como herramienta de desarrollo el Entorno de Desarrollo Integrado (*IDE NetBeans* 7.2). En el informe se detalla el proceso de desarrollo en todas sus etapas: se realiza la planificación y el diseño de la funcionalidad, posteriormente se lleva a cabo la implementación y finalmente se somete a pruebas.

Palabras claves: *Moodle*, actividades, recursos, funcionalidad, Recolector.

Índice de Contenidos

Contenido

Introducción	1
CAPÍTULO 1	5
Introducción.....	5
1.1 El <i>e-learning</i>	5
1.2 Sistema de administración de aprendizaje (LMS)	6
1.3 Plataforma de teleformación <i>Moodle</i>	7
1.3.2 La gestión de un curso <i>Moodle</i>	11
1.3.3 La gestión de un usuario en <i>Moodle</i>	11
1.3.4 Características de la versión 2.0 de <i>Moodle</i>	12
1.4 Estado del arte	14
1.4.1 Conclusiones del estudio realizado	18
1.5 Metodología de desarrollo, lenguajes y herramientas	18
1.5.1 Metodología de desarrollo de <i>software</i>	18
1.5.2 Lenguajes para el desarrollo de la solución.....	20
1.5.3 Herramientas a utilizar en el desarrollo del sistema	23
Conclusiones parciales.....	25
CAPÍTULO 2	26
Introducción.....	26
Diagnóstico inicial.....	26
2.1 Descripción de los procesos vinculados al campo de acción	27
2.1.1 Propuesta del sistema	27
2.2 Principales particularidades de la funcionalidad “Recolector”	28

2.2.1 Funcionalidades del “Recolector”	28
2.2.2 Características del “Recolector”	28
2.2.3 Personal relacionado con la funcionalidad “Recolector”	29
2.3 Fase Exploración.....	29
2.3.1 Historias de usuario.....	30
2.3.2 Estimación	35
2.3.3 Estimación de esfuerzos por Historias de usuario.....	35
2.4 Fase de planificación.....	35
2.4.1 Plan de iteraciones	36
2.4.2 Plan de duración de las iteraciones.....	37
2.4.3 Plan de entregas	37
2.4.4 Diseño del sistema	38
Conclusiones Parciales	41
CAPÍTULO 3.....	43
Introducción.....	43
3.1 Fase de implementación	43
3.1.1 Iteración 1.....	43
3.1.2 Iteración 2.....	44
3.1.3 Iteración 3.....	45
3.1.4 Diagrama de despliegue.....	46
3.2 Fase de Prueba.....	47
3.2.1 Pruebas de aceptación.....	47
3.2.2 Resultados obtenidos de las pruebas.....	53
Conclusiones parciales.....	54
Conclusiones	55

Recomendaciones	56
Bibliografía Referenciada.....	57
Glosario de términos.....	60
Anexos	62

Índice de tabla

Contenido

Tabla 1: Descripción del personal relacionado con la aplicación.	29
Tabla 2: Estructura de una Historia de usuario.	30
Tabla 3: Descripción de la HU, determinar tiempo de inactividad límite.....	31
Tabla 4: Descripción de la HU, listar usuarios pasivos.....	31
Tabla 5: Descripción de la HU, enviar notificación.	32
Tabla 6: Descripción de la HU, eliminar usuarios.....	33
Tabla 7: Descripción de la HU, mostrar actividades y recursos no utilizados.....	33
Tabla 8: Descripción de la HU, enviar notificación a profesores editores de un curso.....	34
Tabla 9: Descripción de la HU, eliminar actividades y recursos no utilizados.	35
Tabla 10: Estimación de esfuerzos por Historias de usuario.....	35
Tabla 11: Distribución de Historias de usuario por iteración.....	37
Tabla 12: Plan de Entrega de la funcionalidad “Recolector”.....	38
Tabla 13: Estructura de las tarjetas CRC.	38
Tabla 14: Descripción de la tarjeta CRC users_listing.....	39
Tabla 15: Descripción de la tarjeta CRC message_post_message.....	39
Tabla 16: Descripción de la tarjeta CRC delete_user.....	39
Tabla 17: Descripción de la tarjeta CRC courses.....	39
Tabla 18: Descripción de la tarjeta CRC fast_modinfo.....	39
Tabla 19: Descripción de la tarjeta CRC delete_course_module.....	40
Tabla 20: Tarea, determinar tiempo límite de inactividad.	44
Tabla 21: Tarea, listar usuarios pasivos.....	44

Tabla 22: Tarea, mostrar actividades y recursos no utilizados.	44
Tabla 23: Tarea, enviar notificación a usuarios pasivos.	45
Tabla 24: Tarea, enviar notificación a profesores editores de un curso.....	45
Tabla 25: Tarea, eliminar usuario.	45
Tabla 26: Tarea, eliminar actividades y recursos inactivos.	46
Tabla 27: Prueba de aceptación 1.....	48
Tabla 28: Prueba de aceptación 2.....	49
Tabla 29: Prueba de aceptación 3.....	50
Tabla 30: Prueba de aceptación 4.....	50
Tabla 31: Prueba de aceptación 5.....	51
Tabla 32: Prueba de aceptación 6.....	52
Tabla 33: Prueba de aceptación 7.....	53
Tabla 34: Registro de defectos y dificultades detectados.....	68

Introducción

El auge de las nuevas Tecnologías de la Información y la Comunicación (TIC) ha tenido gran impacto en la organización de la enseñanza y el proceso de aprendizaje. La acomodación del entorno educativo a este nuevo potencial y la adecuada utilización didáctica del mismo supone un reto sin precedentes. El sistema educativo no puede quedar al margen de los nuevos cambios, debe atender a la formación de los nuevos ciudadanos y la incorporación de las nuevas tecnologías ha de hacerse con la perspectiva de favorecer los aprendizajes y facilitar los medios que sustenten el desarrollo de los conocimientos. (Perez, 2010)

Muchos centros educacionales del mundo y Cuba han comenzado a hacer uso de estas nuevas tecnologías. La introducción de estas en el ámbito educativo ha ayudado en la búsqueda de material didáctico y propiciado el surgimiento del *e-learning* como una alternativa viable para la formación a distancia. Según el Grupo de Trabajo de "e-Learning" 05 de la Red TTNNet España este se define como: "conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet / Intranet, que facilitan el acceso a la información y la comunicación con otros participantes". (Angulo, 2006)

Para contribuir con la inserción del e-learning en la educación se crean herramientas entre las que se pueden mencionar los Sistemas de Gestión del Aprendizaje (del inglés, *Learning Management System*, LMS) y los Sistemas de Gestión de Contenidos del Aprendizaje (del inglés, *Learning Content Management System*, LCMS). Estas automatizan la administración de acciones de formación y controlan el proceso de enseñanza-aprendizaje. Dentro de los LMS de código abierto, uno de los más difundidos es la plataforma de teleformación *Moodle* (del inglés, *Modular Object-Oriented Dynamic Learning Environment*). (Gutiérrez Gómez, y otros, 2007)

El Sistema *Moodle* es una aplicación para crear sitios web dinámicos para el aprendizaje en línea, el cual cuenta con una comunidad de usuarios que tiene como objetivo el desarrollo colaborativo de módulos y personalizaciones de esta plataforma. La plataforma *Moodle* presenta módulos que brindan toda una variedad de opciones entre las que se encuentran: los cuestionarios evaluativos online, los foros de discusión, los registros de evaluaciones, encuestas, tareas, glosarios, lecciones, tareas, talleres y recursos.

Introducción

En la Universidad de las Ciencias Informáticas (UCI) existe una infraestructura creada para la teleformación a partir del Entorno Virtual de Aprendizaje (EVA) *Moodle* como alternativa educativa desde el año 2005, la cual está funcionando en su versión 1.9.12+. Desde su puesta en funcionamiento ha brindado una excelente opción de estudio, constituyendo este un medio donde se recopilan y exponen a la vez todas las materias de los cursos agrupadas por año, además de brindar información y realizar otras acciones que permiten mejorar la calidad de las clases. Su objetivo es proporcionar una mejor interacción profesor-estudiante ayudando a los educadores a crear comunidades de aprendizaje en línea y fomentando el proceso de Enseñanza-Aprendizaje de cada estudiante de forma individual.

En el año 2012 la UCI decide migrar a la versión 2.0 de *Moodle*, la misma posee disímiles características con las cuales no se contaba en las versiones anteriores dentro de las más relevantes se pueden destacar: los *community hubs* estos brindan la posibilidad de crear directorios de cursos públicos para el bien de las comunidades. Mediante los repositorios se pueden integrar los contenidos externos hacia *Moodle* desde otras aplicaciones, con las que interactúa la plataforma y la creación de nuevos bloques.

Actualmente los administradores del EVA refieren que algunos usuarios y cursos existentes en la plataforma dejan de ser utilizados luego de determinado momento y aunque nunca más acceden o son visitados, según sea el caso, perduran. La información como: los usuarios pasivos y actividades o recursos de los cursos inactivos será referida en lo adelante como información no utilizada.

La acumulación de información no utilizada en la plataforma *Moodle* conlleva una utilización innecesaria de espacio en la base de datos y servidores donde se aloja la aplicación. Por otro lado para identificar a los usuarios y cursos inactivos el administrador debe hacerlo de forma muy complicada. Por ejemplo en el caso de los primeros recurriendo a un listado con los usuarios y las fechas de acceso, para luego de calcular el tiempo de inactividad de cada uno decidir qué acciones tomar, las cuales no incluyen una notificación al usuario. Esta dificultad provoca lentitud y complejiza la toma de decisiones por parte de los administradores.

Introducción

Debido a la situación problemática descrita anteriormente se define como **problema a resolver**: ¿Cómo favorecer la gestión por parte de los administradores del sistema de la información no utilizada en la plataforma *Moodle*?

La presente investigación tiene como **objeto de estudio**: El proceso de gestión de la información no utilizada en los LMS.

Para darle solución al problema propuesto se traza como **objetivo general**: Desarrollar una funcionalidad para el módulo de administración de *Moodle 2.3.x* que permita a los administradores del sistema tomar acciones respecto a la información no utilizada.

Se tiene como **campo de acción**: El proceso de gestión de la información no utilizada en el EVA.

Como **idea a defender** se plantea la siguiente: Con la implementación de la funcionalidad “Recolector” para el módulo de administración de *Moodle 2.3.x* se facilitará el control y la toma de decisiones por parte de los administradores de la plataforma sobre la información no utilizada.

Para darle cumplimiento al objetivo se han definido los siguientes **objetivos específicos**:

- Elaborar los referentes teóricos del tema investigado: la gestión de la información no utilizada en los LMS, especialmente *Moodle*, de modo que se avale el problema y se fundamente la propuesta de solución.
- Seleccionar la metodología, lenguajes y herramientas más adecuados para el desarrollo de la funcionalidad “Recolector”.
- Generar los artefactos del análisis y diseño necesarios.
- Implementar la funcionalidad “Recolector”.
- Probar las funcionalidades implementadas para detectar posibles errores.

Los métodos de investigación son el conjunto de procedimientos lógicos que sigue la investigación para descubrir las relaciones internas y externas de los procesos de la realidad natural y social. Para la realización del presente trabajo se hizo necesaria la utilización de los siguientes métodos de investigación:

Métodos teóricos:

Analítico - Sintético: Se utilizó este método para fundamentar el tema y conformar la base teórica de modo que permitió comprender y seleccionar la metodología, lenguajes y herramientas necesarias para lograr una adecuada implementación de la funcionalidad deseada. A partir de un análisis de la bibliografía se arribó a conclusiones que contribuyeron a comprender el problema.

Histórico - Lógico: Este método permitió determinar cómo han evolucionado los sistemas de gestión de cursos y asignaturas, conformar el estado del arte, así como la definición de las características esenciales de la funcionalidad.

Métodos Empíricos:

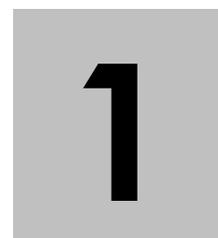
Entrevista: Este método se utilizó para precisar el problema a resolver asociado con las necesidades de los administradores del EVA. Además, para obtener información sobre la gestión de los cursos y usuarios no utilizados en la plataforma.

El documento se ha estructurado en 3 capítulos, los cuales se describen a continuación:

Capítulo 1: Fundamentación Teórica. Se explican conceptos importantes utilizados a lo largo del documento, facilitando así la comprensión de temas posteriores. Además, se describe el estado del arte de las aplicaciones informáticas para la gestión académica. Por último, se detallan la metodología, lenguajes, y herramientas de desarrollo de software que se utilizan en la implementación.

Capítulo 2: Exploración y Planificación. Se expone la fase exploración y planificación de la metodología XP, se muestran los resultados obtenidos en el flujo de trabajo de diseño del sistema; los cuales serán usados posteriormente en la implementación del “Recolector”. En este flujo de trabajo se realiza un refinamiento del análisis, teniendo en cuenta las funcionalidades y características del sistema, garantizando que la funcionalidad cumpla con los requerimientos deseados, considerando además el entorno de implementación.

Capítulo 3: Implementación y Prueba. Se expone el flujo de trabajo de implementación, además del diagrama de despliegue a partir de lo obtenido en el diseño, por último se exponen los resultados de las pruebas realizadas al sistema.



CAPÍTULO

Introducción

En este capítulo se presentan los fundamentos teóricos de la investigación. Se determina el concepto de *e-learning* que se manejará durante el resto del informe. Se describe la estructura y los conceptos relacionados con la plataforma *Moodle*. Se presentan los resultados del estudio realizado a módulos con características similares al que se pretende implementar y por último, se describen la metodología, los lenguajes y herramientas a utilizar durante el proceso de desarrollo.

1.1 El *e-learning*

La integración de las TIC a los procesos educativos como parte de esta revolución tecnológica provocó el surgimiento del aprendizaje electrónico o e-Learning usándose varias definiciones para describirlo. Esta modalidad educativa utiliza Internet siendo un nuevo paso en el tema de la educación a distancia permitiendo que los estudiantes puedan comunicarse y colaborar con sus compañeros y docentes (profesores, tutores y mentores), de forma síncrona o asíncrona, sin limitaciones espacio-temporales. Debido a su impacto, catedráticos, profesionales y especialistas en el tema se han incitado a emitir diversos conceptos al respecto, los cuales se muestran a continuación.

“El *e-learning* es el suministro de programas educacionales y sistemas de aprendizaje a través de medios electrónicos, abarca un amplio paquete de aplicaciones y procesos, como el aprendizaje basado en Web, capacitación basada en computadoras, salones de clases virtuales y colaboración digital (trabajo en grupo)”. (Ruiz Ortiz , y otros, 2011)

Por otra parte, este término consiste en un: “amplio grupo de aplicaciones y procesos, tales como: aprendizaje basado en web, aprendizaje basado en ordenadores, aulas virtuales y colaboración digital. Incluye entrega de contenidos vía Internet, intranet/extranet, audio y video grabaciones, transmisiones satelitales, TV interactiva, CD-ROM y más”. (Presso, 2011)

Una vez estudiadas cada una de las definiciones anteriores, la autora determina utilizar para el desarrollo de la presente investigación la siguiente definición de *e-learning*: “conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes”. (Angulo, 2006)

E-learning brinda una gran variedad de ventajas que facilitan el proceso de enseñanza-aprendizaje entre las cuales se pueden encontrar las siguientes: (Accogli, 2006)

- Permite una mayor flexibilidad en cuanto al tiempo de estudio haciendo más responsable al estudiante y más profesional.
- Permite un seguimiento más completo y riguroso del estudiante en cuanto a sus tareas docentes.
- Promueve la autonomía de los participantes y fluye el intercambio de contenidos a través del correo electrónico, foros de discusión, videoconferencias, entre otros.
- Brinda una mayor facilidad en cuanto al acceso y actualización de contenidos.
- Permite interactuar ya sea profesor-alumno, de manera grupal o trabajar individual.

La demanda y rápida difusión del aprendizaje electrónico, potenció la creación de herramientas electrónicas que brindaran soporte al nuevo método. Entre estas aplicaciones se encuentran los Sistemas Gestores del Aprendizaje (LMS).

1.2 Sistema de administración de aprendizaje (LMS)

Un LMS es una aplicación residente en un servidor en la que se desarrollan las acciones formativas. Es sinónimo de plataforma o campus virtual. Es el lugar donde alumnos, tutores, profesores o coordinadores se conectan a través de Internet para descargar contenidos, ver programas de asignaturas, enviar correos al profesor, charlar con los compañeros, debatir en un foro, participar en una tutoría, entre otros. (Gutiérrez Gómez, y otros, 2007)

También se plantea que un LMS es un sistema enfocado al área educativa que automatiza la administración no solo de acciones de formación y de los distintos usuarios que interactúan con este, sino también de los contenidos que proporciona. Los LMS tienen la capacidad de integrar personas, procesos y equipos en aulas virtuales para el

aprendizaje en línea. Publicado en el artículo “Los estándares de *e-Learning*” de Carlos Emilio Biscay¹, en el sitio oficial de la Universidad de Palermo. (Biscay, 2010)

Dentro de los LMS más difundidos, se encuentra *Moodle*, una plataforma educativa que es utilizada a nivel internacional por las ventajas que brinda a los educadores.

1.3 Plataforma de teleformación *Moodle*

En este epígrafe se definirá el concepto de *Moodle* y las principales ventajas de la última versión (2.0) de esta plataforma. También se describirán los diferentes bloques que posee y se detallará el bloque en el que se encuentra ubicada la funcionalidad “Recolector”. Además, se describe el proceso de gestión de la información no utilizada en el EVA de la UCI actualmente.

El Entorno de Aprendizaje Dinámico Modular Orientado a Objetos (*Moodle*) es una plataforma para la creación de cursos y sitios web basados en Internet. Tiene como misión principal desarrollar nuevas teorías educativas basadas en toda una gama de recursos didácticos disponibles en ella. *Moodle* es muy utilizado en varios centros de enseñanza, constituyendo una herramienta muy útil para estos propósitos, pues soporta disímiles contenidos como son textos, animaciones, videos, imágenes, entre otros.

Este sistema se distribuye gratuitamente como *software* libre y funciona en cualquier computadora en la que pueda correr PHP. Soporta varios tipos de bases de datos, esencialmente *MySQL*. Asegura un alto rendimiento combinando una buena elección del hardware, sistema operativo y sistema de bases de datos. (Pupo, 2011)

Moodle posee varios módulos o bloques en los que se organizan las diferentes funcionalidades del sistema. En el siguiente epígrafe se describen los bloques más importantes dentro de la plataforma según su usabilidad y su relación con otros módulos.

1.3.1 Bloques de la plataforma de teleformación *Moodle*

La plataforma de teleformación *Moodle* posee bloques estándares y no estándares. Los bloques no estándares son opcionales, los cuales deben ser descargados e instalados

¹ Carlos Emilio Biscay, licenciado en Sistemas (CAECE). Ha realizado estudios de posgrado en estadística aplicada a la investigación humana UBA-Universidad Libre de Bruselas y actualmente está completando un master en MBA (Orientación Management) en la Universidad de Palermo.

previamente, mientras que los estándares vienen incluidos por defecto en la plataforma de teleformación *Moodle*.

Dentro de este entorno se pueden configurar, instalar y desinstalar los bloques. Estos encierran un conjunto de funcionalidades que robustecen el sistema. Se encuentran en ambos lados de la pantalla, dejando el centro para visualizar los contenidos del sitio. Con los permisos de administración, estos pueden ser ocultados, suprimidos y desplazados lateral o verticalmente.

Existen varios tipos de bloques estándares con funcionalidades muy diversas, algunos trabajan de forma independiente, otros colaboran entre sí. A continuación se describen los más relevantes: (Torre, 2006)

1. **Bloque “Actividad reciente”:** Muestra los últimos movimientos de un usuario dentro del curso. Entre otras cosas, el alumno podrá observar sus últimas participaciones en los foros de la asignatura o las actividades programadas.
2. **Bloque “Actividades”:** Permite al usuario un acceso rápido a los diferentes tipos de actividades incluidas en un curso.
3. **Bloque “Administración”:** Mediante este bloque el usuario puede acceder a diferentes herramientas administrativas en función de su perfil. Desde estas opciones el alumno puede examinar sus calificaciones, cambiar su contraseña o darse de baja de un curso (el profesor se encargará de decidir que opciones estarán disponibles para sus alumnos). Por otra parte, el profesor puede configurar el aspecto de un curso (incorporando bloques por ejemplo), añadir nuevas actividades o hacer copias de seguridad. Cuanto mayor el nivel de privilegios, de más opciones se podrá disponer. Este bloque cuenta con varias opciones entre las que se encuentran:
 1. **Notificaciones:** Permite al administrador del sitio tener siempre presente la versión de *Moodle* con la que está trabajando.
 2. **Usuarios:** Este menú cuenta con dos funcionalidades.
 - **Cuentas:** Esta incluye, hojear lista de usuarios, acciones de usuario masivas la cual permite, agregar usuario, subir usuarios, insertar los usuarios mediante un archivo, subir imágenes de los usuarios y campos de perfil del usuario.

- Permisos: Brinda la posibilidad al administrador del sitio de definir los roles en los cursos, asignar los roles globales y tratar políticas de usuario.
3. **Cursos:** Permite configurar los cursos y sus categorías, además crear otros nuevos, gestionar los existentes o realizar copias de seguridad de todos los que ya existen en la plataforma.
 4. **Calificaciones:** Permite realizar ajustes generales de las calificaciones, y de todos los elementos relacionados con esta categoría, como escalas, apariencia o la representación de los resultados.
 5. **Ubicación:** Permite modificar la ubicación de la zona horaria.
 6. **Seguridad:** Permite configurar el sistema de seguridad de todo el sitio o sólo de los módulos y bloques, también se puede usar para activar el antivirus.
 7. **Servidor:** Contiene enlaces que permiten configurar el servidor que soporta la plataforma *Moodle*, así como la configuración de todo el sistema, de correo electrónico o de estadísticas, también permite a través de la opción limpieza fijar el tiempo de caducidad para cuentas de usuario, las cuentas no confirmadas y los registros de acceso y uso, por este motivo se decidió ubicar en este menú la funcionalidad “Recolector”.
4. **Bloque “Cursos”:** Muestra al usuario todos los cursos en los cuales participa ordenados por categoría. Esto le permitirá desplazarse cómodamente entre sus cursos.
 5. **Bloque “Eventos Próximos”:** Recuerda la proximidad de un evento, ya sea un evento general, a nivel de curso o programado por el propio usuario. Los eventos se programan en el bloque Calendario, donde también, se puede configurar el número de eventos que se mostrarán en este bloque o la fecha a partir de la cual un evento puede ser considerado como próximo.
 6. **Bloque “Personas”:** El bloque “Personas” agrupa opciones relacionadas con la información recogida por *Moodle* sobre los participantes de un curso, bien sean alumnos o profesores. Por ejemplo, se puede visualizar la información del resto de alumnos, a qué grupo pertenecen o modificar los datos de la ficha personal.
 7. **Bloque “Usuarios en línea”:** Muestra los usuarios conectados en línea en un curso. Se detectan nuevos usuarios pasado un tiempo (en minutos) configurable por el administrador.

Entre los bloques no estándares que se han desarrollado como contribución para la plataforma, se pueden mencionar los bloques C2Site (*Course To Site*) y C2SCORM (*Course To SCORM*) desarrollados por el proyecto Personalizaciones de Moodle de la UCI, los cuales permiten la exportación de contenidos.

1. **Bloque C2Site:** Permite exportar el contenido de los cursos, que posean un formato por semanas o por temas, en forma de sitio web estático, brindándole la posibilidad al usuario de utilizar los cursos en entornos sin conexión de red. Dicho sitio puede contener uno o varios cursos, según lo estime el usuario al cual se le hayan dado los permisos necesarios para exportarlos. Entre sus principales funcionalidades se encuentran las siguientes:

- Exportar curso: Permite exportar los cursos de la plataforma de teleformación *Moodle* estructurado por temas o por semanas en forma sitio web estático y almacenarlos en un directorio raíz seleccionado por el usuario.
- Exportar un curso con otros ya exportados: Permite exportar un curso dentro de uno que ya haya sido exportado.
- Administrar los cursos exportados: Permite realizar acciones como copiar, mover, eliminar, ver, renombrar o compactar un paquete de curso.

2. **Bloque C2SCORM:** Permite exportar el contenido de los cursos, con formato por temas o por semanas, en formato de paquete SCORM, con la finalidad de compartir, reutilizar y exportar los contenidos de *Moodle* a cualquier otra plataforma que sea compatible con este formato. El bloque realiza la exportación de contenidos bajo el estándar SCORM 1.2 o SCORM 2004, dependiendo de las versiones desarrolladas. Para la descripción de contenidos hace uso del estándar LOM versión 1.0. Entre sus principales funcionalidades se encuentran las siguientes:

- Exportar un curso de *Moodle* posibilitando seleccionar los recursos y actividades involucrados en el proceso.
- Permitir el llenado manual de metadatos a los recursos.
- Configurar el proceso de exportación.
- Descargar el paquete previamente exportado. (Tarruella, 2010)

1.3.2 La gestión de un curso Moodle

Un curso en *Moodle* es un área donde un profesor puede agregar recursos y actividades para sus alumnos. Esto podría ser una simple página con documentos descargables o podría ser un complejo conjunto de tareas en las que el aprendizaje progresa a través de la interacción.

La página del curso se compone de secciones centrales que contengan las tareas y los bloques (si se desea) a un lado. El profesor del curso tiene el control sobre el diseño de la página principal del curso y puede cambiar en cualquier momento. Los estudiantes pueden inscribirse de forma manual por parte del profesor, de forma automática por el administrador, o bien se puede permitir que se inscriba. Los estudiantes también se pueden añadir a los grupos si tienen que ser separados de clases que comparten el mismo curso o si las tareas deben ser diferenciadas. (Pech, 2008)

Mediante la gestión de un curso no se puede conocer las actividades y recursos que no están siendo utilizados en cada uno de los cursos disponibles en la plataforma. Cuando un administrador crea un curso debe proceder a su configuración mediante un formulario proporcionado por *Moodle* en el que se establecen valores para distintos campos como, por ejemplo:

- Nombre y descripción.
- Formato del curso (semanal, por temas)
- Número de semanas o temas.
- Fechas en las que permanecerá abierto el curso.

1.3.3 La gestión de un usuario en Moodle

Un usuario es un participante que tiene permisos en un sitio *Moodle*. Versiones previas de la plataforma asignaban sencillos roles para definir los permisos de usuario. A medida que *Moodle* ha ido madurando, estos permisos se han especializado cada vez más mediante su uso en distintos contextos. Los roles de usuario habituales son: administradores, profesores, estudiantes, invitados y creador de cursos. (*Moodle*, 2007)

1. Estudiantes: Es el tipo de rol más básico de *Moodle*, son los alumnos en cada curso.

2. Creador de cursos: La creación de nuevos cursos sólo puede ser realizada por un usuario que tenga asignado ese privilegio, además permite asignar profesores y actuar como profesor con privilegios de edición. El creador del curso puede ser el profesor principal, el jefe de departamento o el coordinador del programa.
3. Profesores: Tienen el control sobre un curso específico dentro de *Moodle* y la actividad de los alumnos que están inscritos en él.
4. Invitados: Tienen acceso de “solo-lectura”, lo que significa que no pueden enviar mensajes o entorpecer el curso de los verdaderos alumnos. Estos no pueden:
 - Enviar mensajes a discusiones.
 - Editar páginas Wiki.
 - Contestar cuestionarios.
 - Enviar tareas
 - Contribuir al contenido o comentarios de los glosarios.
5. Administrador: Un administrador en *Moodle* gestiona todo el sitio. Normalmente supervisa la apariencia y la sensación que produce el *Moodle* de su organización y que lo hace único. Este rol o categoría ocupa el nivel más alto en *Moodle*, en lo que a privilegios de usuarios se refiere. Los privilegios de usuarios permiten restringir ciertas características o funcionalidades a un usuario en función de su nivel. Es el encargado de crear, administrar los demás usuarios. Existen las siguientes categorías de usuarios, ordenadas de mayor a menor en función del nivel de privilegios:
 - Administrador principal.
 - Administrador.
 - Creador de cursos.
 - Profesor editor.
 - Profesor no editor.
 - Alumno e invitado.

1.3.4 Características de la versión 2.0 de *Moodle*

Muchas han sido las versiones liberadas por la comunidad de *Moodle*, a partir de la versión 2.0, todas las desarrolladas hasta llegar a la 2.3.x poseen nuevas características, pues se llevó a cabo prácticamente todo un nuevo desarrollo de la plataforma. Dentro de

los nuevos elementos incorporados se pueden destacar: los *community hubs*²; brindan la posibilidad de crear directorios de cursos públicos para el bien de las comunidades. Por otro lado mediante los repositorios se pueden integrar los contenidos externos hacia *Moodle* desde otras aplicaciones con las que interactúa la plataforma, como es el caso de *Mahara*³, *Flirck11*⁴, entre otros.

A partir de la versión 2.0 se adicionaron a *Moodle* nuevas características a continuación se describen las principales:

Incorporación de nuevos bloques:

- Bloque de comentarios: Permite comentarios que pueden ser agregados a cualquier página. Ideal para opinión de estudiantes.
- Mi bloque de archivos privados: Permite fácil acceso a los archivos propios en un repositorio de archivos privado en *Moodle* (con soporte de cuotas).
- Bloque de comunidad: Mantiene registro de cursos externos en los que un usuario está interesado.
- Bloque de estado de completado de curso: Reporta el estado de avance de los cursos de un usuario. (*Moodle*, 2011)

Cursos

- Los profesores ahora pueden especificar condiciones estándar para todos los estudiantes. Las condiciones pueden ser de completado de actividades, pero también condiciones por calificación, fecha y otros criterios.
- Los profesores pueden utilizar las condiciones como prerequisites para acceder a otros cursos, lo que permite una progresión ordenada y la construcción de itinerarios.
- Los profesores y estudiantes pueden ver informes que muestran el progreso dentro del curso, o a través de una serie de cursos.

² **Community hubs**, consisten en directorios de cursos para uso público o comunidades privadas.

³ **Mahara**, es una aplicación web en código abierto para gestionar ePortfolio y Redes sociales. Ofrece a los usuarios herramientas para crear y mantener un portafolio digital sobre su formación.

⁴ **Flickr**, es un sitio web gratuito que permite almacenar, ordenar, buscar, vender y compartir fotografías y videos en línea.

Roles y permisos

- Simplificada la lógica de evaluación de los permisos.
- Interfaz mejorada y simplificada en AJAX para definir y asignar roles.
- Interfaces mejoradas y simplificadas para controlar permisos en cualquier contexto dado.
- Nuevo concepto de "Arquetipos" reemplazando el concepto de "Legacy roles".
- Nuevo arquetipo "manager" para definir el rol de mayoría de personas con privilegios de edición a lo largo de la plataforma, separado del rol "admin".
- Los permisos de los superusuarios de tipo "Administrador" no pueden ser modificados.

Requerimientos

- PHP 5.2.8 o superior, PHP 5.3.3 o posterior es recomendado.
- Bases de datos deben ser alguna de estas:
 1. MySQL 5.0.25 o superior (usar motor de almacenamiento InnoDB es altamente recomendado).
 2. PostgreSQL 8.3 o superior.
 3. Oracle 10.2 o superior.
 4. MySQL 2005 o superior.
- Cualquier navegador de Internet que soporte estándares de los últimos años, por ejemplo:
 1. Firefox 3 o superior.
 2. Safari 3 o superior.
 3. Google Chrome 4 o superior.
 4. Opera 9 o superior.
 5. MS Internet Explorer 7 o superior.

1.4 Estado del arte

En la búsqueda de una respuesta a las necesidades actuales de los administradores del EVA se estudiaron diferentes Sistemas de Gestión de Aprendizaje con el objetivo de

tomar experiencias en cuanto al funcionamiento de estos módulos y al mismo tiempo analizar en qué medida brindaban solución a la problemática planteada.

Los sistemas que se tuvieron en cuenta para el análisis realizado cumplen con ciertas características, son sistemas publicados bajo los conceptos del software libre y similares a *Moodle* en cuanto al manejo de la información. Hecho por el cual solamente se describen 2 de entre los muchos LMS existentes.

Sistemas de Gestión de Aprendizaje *Claroline* es un sistema de gestión de aprendizaje (LMS), originalmente desarrollado en el 2001 por Thomas De Praetere en la Universidad Católica de Louvain en el IPM (Institut de Pédagogie universitaire et des Multimédias, Bélgica), actualmente es fruto de la colaboración entre varias instituciones y desarrolladores de distintas partes del mundo.

Plataforma de aprendizaje y trabajo virtual (*eLearning* y *eWorking*) de código abierto y *software* libre (*open source*) que posibilita a los docentes construir eficaces cursos en línea (*online*) y gestionar las actividades de aprendizaje y colaboración en la web, tales como: (Romani, 2012)

- Publicar documentos en cualquier formato, ejemplo word (doc), pdf, html y video.
- Administrar foros de discusión tanto públicos como privados.
- Administrar listas de enlaces.
- Crear grupos de estudiantes.
- Confeccionar ejercicios.
- Estructurar una agenda con tareas y plazos.
- Hacer anuncios, vía correo electrónico por ejemplo.
- Gestionar los envíos de los estudiantes.
- Crear y guardar chats.

Esta plataforma permite a los profesores crear y administrar los cursos desde un navegador (*Explorer*⁵, *Mozilla*⁶ y *Netscape*⁷), además brinda la posibilidad de modificar, eliminar, y controlar el acceso a dicho curso. *Claroline* también administra y agrega los

⁵ **Windows Internet Explorer**, conocido comúnmente como IE, es un navegador web desarrollado por Microsoft para el sistema operativo Microsoft Windows desde 1995.

⁶ **Mozilla Firefox**, es un navegador web libre y de código abierto desarrollado para Microsoft Windows, Mac OS X y GNU/Linux coordinado por la Corporación Mozilla y la Fundación Mozilla.

⁷ **Netscape Navigator**, fue un navegador web y el primer producto comercial de la compañía Netscape Communications, creada por Marc Andreessen, uno de los autores de Mosaic.

Fundamentación Teórica

usuarios en la plataforma, la herramienta Usuarios provee una lista que muestra el nombre, apellidos, perfil, rol, y el grupo al que pertenece cada estudiante inscrito en el curso, además posee la opción seguimiento de un usuario la cual permite al profesor conocer las estadísticas sobre las actividades que ha realizado un usuario particular en el curso, los usuarios pueden ser de 3 clases:

- Estudiante: Este es un usuario que puede tener acceso únicamente a los materiales y herramientas de los cursos creados por un profesor.
- Profesor: Esta clase de usuario tiene todos los derechos del estudiante y además tiene los derechos de crear cursos nuevos y de modificar todos sus contenidos. Puede organizar los estudiantes inscritos en su curso en grupos de trabajo y visualizar las actividades que ha realizado cada estudiante de su curso.
- Administrador: Es la clase de usuario que tiene los derechos más amplios sobre la plataforma. Puede asignar derechos especiales a todas las clases de usuarios y es el que normalmente registra a los profesores.

El LMS *Claroline* es muy semejante a *Moodle* pues posee una opción para crear y administrar cursos permitiendo a los administradores eliminarlos por completo y otra para agregar usuarios en la plataforma. A pesar de esto, posee el inconveniente de que no cuenta con una funcionalidad que permita conocer el estado de los usuarios, o sea saber si están inactivos en la plataforma, ni conocer las actividades y recursos que no son utilizados dentro de los cursos.

Sistemas de Gestión de Aprendizaje Dokeos es una plataforma de *e-learning*, que permite a los docentes y alumnos las funciones administrativas y académicas de la capacitación, el desarrollo es internacional y colaborativo este posee: (Favier, 2010)

- Herramienta para la creación de contenido propia.
- Baúl de tareas (*Dropbox*): posibilidad de intercambiar documentos entre alumnos y con el profesor de manera privada.
- Agenda.
- Mejoramiento de la seguridad: encriptación de claves.
- Interoperabilidad de base de datos: XML, CSV, Excel, Oracle, SQL-Server, *FileMaker*.
- Vista del estudiante y del profesor.

Fundamentación Teórica

El desarrollo de *Dokeos* es un proyecto internacional que incluye como contribuyentes a varias universidades, escuelas, y otras organizaciones e individuos. *Dokeos* permite que los profesores puedan acceder con facilidad a las prestaciones del sistema, administrar sus documentos en procesador de texto, pdf, gráficos y planilla de cálculos, sin necesidad de efectuarles ningún tratamiento informático especial. El sistema cuenta con un asistente para elaborar el programa del curso o materia, el cual brinda la opción de agregar información sobre el curso, objetivos y programa de estudios.

Mediante una serie de preguntas y recomendaciones por pantalla se va guiando al profesor para permitirle hacer o perfeccionar la comunicación de su programa de estudio a la comunidad educativa, también permite Activar / Desactivar los componentes del curso. Este posee varias secciones entre las que se encuentra: “Usuarios”, esta muestra la lista de los cursantes, donde puede controlar quién está registrado en su curso. Los profesores no pueden modificar la información del estudiante como el nombre o contraseña porque depende de ellos actualizar su propio perfil, los usuarios pueden ser de 5 tipos:

- Invitado: son los usuarios que se registran en el sitio universitario con un formulario básico y al ingresar en esta opción les es permitido navegar por la versión demostración del sitio virtual.
- Estudiante: tiene permitido el ingreso a todos los elementos pertinentes especificados. No podrá acceder a la administración de las bases de datos que sustentan el sistema. Ese tipo de información solo está disponible para los profesores, tutores y administrativos.
- Profesor: tiene permitido el ingreso a los elementos pertinentes a su función como por ejemplo: materias que dicta, planillas de inscripción, publicar noticias en las diferentes carteleras y proponer foros de opinión.
- Tutor: tiene permitido el ingreso a los mismos elementos del estudiante y a la mayoría de los elementos del docente; además, del acceso a las materias a las que se encuentra inscripto como tutor. Puede también publicar noticias y proponer foros de opinión. Es un facilitador o guía del alumno.
- Administrador: El administrador del sitio virtual tiene acceso completo a todos los elementos existentes.

Este LMS entre sus opciones no cuenta con ninguna funcionalidad que permita la gestión de la información que no se utiliza. Tampoco brinda opciones para conocer el estado de los usuarios registrados en el sistema. Por otro lado es imposible constatar el total de cursos existentes o sus características, a menos que se haga directamente en la base de datos.

1.4.1 Conclusiones del estudio realizado

Los sistemas encontrados resultan insuficientes para suplir las exigencias del producto requerido por los administradores del EVA de la UCI. En los mismos se encuentran solamente fragmentos de las funcionalidades del producto que se pretende elaborar. Las características de estos LMS aunque con similitudes en algunos casos, no se corresponden con la totalidad de las que se desea incluir en la funcionalidad “Recolector” para *Moodle* 2.3.x. Por tanto, se decide comenzar el desarrollo de una nueva funcionalidad, implementada para responder a los requerimientos y necesidades específicos de los administradores del sistema de la Universidad.

En la implementación de la funcionalidad el “Recolector” se ha de utilizar una metodología de desarrollo, así como diferentes herramientas y lenguajes, ya sean de modelado o programación. En los epígrafes posteriores se describen la metodología, herramientas y lenguajes seleccionados para el desarrollo de la propuesta de solución.

1.5 Metodología de desarrollo, lenguajes y herramientas

En el presente epígrafe se describirá la Metodología de desarrollo, lenguajes y herramientas seleccionados para guiar la implementación de la funcionalidad “Recolector”. La selección parte de los resultados de estudios comparativos realizados anteriormente en tesis de grado (Ver anexo 1), resultado del trabajo de investigación de estudiantes y profesores del Centro de Tecnologías para la Formación (FORTES) de la Universidad de las Ciencias Informáticas.

1.5.1 Metodología de desarrollo de *software*

Un proceso de *software* detallado y completo suele denominarse Metodología, define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Las metodologías se desarrollan con el objetivo de dar solución a los problemas existentes en la producción de *software*, que cada vez son más complejos. Estas engloban procedimientos, técnicas, documentación y herramientas que se utilizan en la creación de un producto de *software*.

No existe una metodología de *software* universal. Las características de cada proyecto exigen que el proceso sea configurable. (Jacobson, y otros, 2000)

Existen varias metodologías de desarrollo de *software* las cuales se dividen fundamentalmente en dos grandes grupos: Metodologías tradicionales y Metodologías Ágiles o Livianas. Según Pressman las metodologías ágiles “...buscan la satisfacción del cliente y la entrega temprana de *software* incremental; equipos de proyectos pequeños y con alta motivación; métodos informales; un mínimo de productos de trabajo de la ingeniería del *software* y una simplicidad general del desarrollo...”. Su objetivo es esbozar los valores y principios que deberían permitir a los equipos desarrollar *software* rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. (Pressman, 2009)

Las características de la funcionalidad “Recolector” para *Moodle* en su versión 2.3.x se ajustan al desarrollo ágil del *software* pues esta será implementada por una persona en 9 semanas y no es un desarrollo complejo. A continuación se describe la metodología de desarrollo seleccionada para la propuesta de solución:

Extreme Programming (XP)

Esta metodología de desarrollo de *software* es una de las más exitosas utilizada en la actualidad para proyectos de corto plazo y con escaso personal. Consiste en una programación rápida o extrema con mucha reutilización de código que ha sido desarrollado previamente. Una particularidad de ella es tener como parte del equipo al usuario final, pues es uno de los requisitos para obtener un resultado satisfactorio.

Esta metodología de desarrollo genera varios artefactos entre los cuales se encuentran las Historias de usuario y las Tarjetas CRC (Clase-Responsabilidad-Colaborador), los mismos se describen a continuación:

Según Kent Beck ⁸el primer paso de cualquier proyecto que siga la metodología X.P es definir las Historias de usuario con el cliente. Las **Historias de usuario** tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles;

⁸ **Kent Beck**, es ingeniero de software estadounidense, uno de los creadores de las metodologías de desarrollo de software de programación extrema y el desarrollo guiado por pruebas, también llamadas metodologías ágiles.

no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados.

Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la Historia de usuario. Cuando llega la hora de implementar una Historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una Historia de usuario es entre 1 y 3 semanas. (Beck, 2012)

Tarjetas CRC: Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores.

1.5.2 Lenguajes para el desarrollo de la solución

Lenguaje Unificado de Modelado (UML) 2.0

El UML es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas.

UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos." (Jacobson, y otros, 2004)

La plataforma de teleformación *Moodle*, sobre la cual se desarrolla la presente investigación, es una aplicación web escrita con el lenguaje script PHP y que utiliza una base de datos SQL para almacenar información. Como sistema modular está estructurado por un núcleo y extensiones que proveen funcionalidades específicas, estas últimas incluyen archivos CCS y *JavaScript*. La información mostrada por *Moodle* debe acatar las restricciones del estándar XHTML 1.0. Es por esto que para el desarrollo de la

solución del problema identificado en la presente investigación son seleccionados los lenguajes descritos a continuación.

Lenguajes de programación del lado del servidor

Los lenguajes del lado del servidor proporcionan un entorno rápido de creación de scripts y soporte para los estándares más importantes, posibilitando la integración con bases de datos. No tienen en cuenta el tipo de cliente; la aplicación se ejecuta en el servidor que es un ambiente controlado, una vez ejecutada la aplicación, el resultado que se envía al cliente puede estar en un formato estandarizado que cualquier cliente puede interpretar. (Negrín, y otros, 2011)

Preprocessed Hypertext Pages (PHP) 5.3.8

Preprocessed Hypertext Pages (PHP) es un lenguaje interpretado de programación, diseñado para desarrollar páginas web dinámicas y de forma rápida, el cual puede ser impregnado en páginas HTML. Es de alto nivel y como es ejecutado desde el servidor, el cliente recibe solamente el resultado buscado por el código en el que está escrito. (Achour, 1997-2013)

Por su naturaleza de código abierto se muestra como una alternativa de fácil acceso al *software* privativo. Es un lenguaje sencillo para el aprendizaje, a partir de las últimas versiones pasó a ser un lenguaje orientado a objetos y es capaz de conectarse con la mayoría de los gestores de bases de datos que se utilizan en la actualidad. (Negrín, y otros, 2011)

Lo que distingue a PHP, es que el código se ejecuta en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá los resultados de ejecutar el script, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los archivos HTML. (Achour, 1997-2013)

Lenguajes de programación del lado del cliente

Los lenguajes del lado del cliente son ejecutados en el navegador del usuario el cual soporta la carga de procesamiento y funcionamiento. Las páginas clientes dependen, por tanto, de las características del sistema en el que se ejecutan.

JavaScript

Es un lenguaje interpretado basado en guiones que son integrados directamente en el código HTML. Es utilizado en el desarrollo web, el código es transferido al cliente para que este lo interprete al cargar la página. Con este lenguaje no pueden crearse programas independientes. Permite agregar animaciones y efectos dotando de dinamismo a la página web al interactuar con el navegador. (Sáenz, 2011)

Entre las características de JavaScript más sobresalientes se encuentran:

- Es dinámico, responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo. Con esto se puede cambiar totalmente el aspecto de la página al gusto del usuario, evitándose tener en el servidor un página para cada gusto.
- Tiene la ventaja de ser incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado.
- Maneja objetos dentro de una página Web y sobre ese objeto se pueden definir diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario, tales como formateo de unidades y modificación de archivos.

Lenguaje de marcado de hipertexto extendido (XHTML) 1.0

Extensible HyperText Markup Language (XHTML por sus siglas en inglés) es una versión más estricta y limpia de HTML, que nace con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos. Este lenguaje es una vuelta hacia atrás, que intenta recuperar la línea marcada por los estándares, pero que trata de solucionar diversos casos de uso del HTML; a la vez que lo prepara para adaptarse a las nuevas necesidades y corrientes tecnológicas. (Clark, 2012)

XHTML no es solo un HTML con sintaxis XML. En realidad, incorpora una nueva concepción o si se prefiere, una nueva filosofía de modelación de las páginas web, que

busca la creación de una web semántica. La web semántica permite separar el contenido de la presentación y eso es algo que se lleva intentando desde hace años y un asunto que se ha puesto en práctica de manera especial en el XHTML.

XHTML procura acercarse a ese ideal, por lo que se suprimen todas las etiquetas y atributos que sirven para definir el aspecto y solo se dejan las etiquetas que sirven para definir el significado de cada elemento de la página. (Álvarez, 2011)

Hojas de Estilo en Cascada (CSS) 2.0

Cascading Style Sheets (CSS por sus siglas en inglés), considerado lenguaje de especificación, es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Funciona a base de reglas, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.

1.5.3 Herramientas a utilizar en el desarrollo del sistema

Herramienta CASE: Visual Paradigm for UML 8.0

La palabra **CASE**⁹ son las siglas en inglés de **Computer Aided Software Engineering** o Ingeniería de *Software* Asistida por Computadora. Una herramienta *CASE* no es más que una aplicación informática que brinda ayuda en el desarrollo de casi todo el ciclo de vida del proyecto. Permite aumentar la productividad reduciendo parámetros como el costo y el tiempo y asiste diversas tareas como la realización del diseño, la generación de código a partir del mismo y la documentación y detección de errores.

⁹ Las herramientas **CASE** (**Computer Aided Software Engineering**, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software.

El desarrollo de la propuesta de solución planteada en la presente investigación será realizado sobre el sistema operativo (SO) *Linux*. Se decide utilizar *Visual Paradigm* para la modelación del *software* pues es una potente herramienta que además es multiplataforma. *Visual Paradigm* es una herramienta de modelado visual para todo tipo de diagramas UML y Diagramas de entidad-relación (DER) para el diseño de bases de datos. (Pressman, 2002)

Presenta las siguientes características:

- Generación de código - Modelo a código, diagrama a código.
- Diagramas de flujo de datos.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas.
- Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XML.

Entorno de Desarrollo Integrado (IDE) NetBeans 7.2

El IDE NetBeans es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. *Sun Microsystems*¹⁰ fundó el proyecto de código abierto *NetBeans* en junio 2000 y continúa siendo el patrocinador principal de los proyectos.

Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además, un número importante de módulos para extender el *NetBeans* IDE. El cual es un producto libre y gratuito sin restricciones de uso. Los desarrolladores pueden crear rápidamente sitios web, utilizando la plataforma PHP.

Este IDE aporta mejoras en cuanto a rendimiento con respecto a otras plataformas y cuenta con un completamiento de código superior a otros entornos, lo que economiza el tiempo de los desarrolladores. Por las características antes expuestas y por ser una herramienta libre, que ofrece una amplia documentación se selecciona *NetBeans* para implementar la solución de la presente investigación.

Gestor de Bases de Datos

¹⁰ **Sun Microsystems**, fue una empresa informática que se dedicaba a vender estaciones de trabajo, servidores, componentes informáticos, software (sistemas operativos) y servicios informáticos.

La plataforma de teleformación Moodle es independiente de los sistemas gestores de base de datos que soporta, siendo estos: MySQL, PostgreSQL, MSSQL y Oracle. Para establecer la conexión con el sistema de base de datos se utiliza el protocolo ADO, el mismo es una interfaz de acceso a datos, que permite que los programas cliente obtengan acceso, para consultar y modificar la información contenida en diferentes áreas de almacenamiento dentro de la base de datos llamadas tablas. Además, se puede manipular la propia base de datos para crear nuevas áreas para el almacenamiento de información (tablas), como también alterar o eliminar las ya existentes.

Conclusiones parciales

Al analizar la forma en que se gestionan los cursos y usuarios de la plataforma de teleformación de la UCI, las principales actividades llevadas a cabo por los administradores y estudiar las características de algunos LMS existentes, se puede concluir que no existe ninguna funcionalidad capaz de gestionar la información no utilizada en *Moodle*, por lo tanto se decidió comenzar la implementación de la misma.

Como metodología de desarrollo de *software* se decide utilizar XP, asistida por la herramienta *CASE Visual Paradigm 8.0*, pues es la más adecuada para guiar el proceso de desarrollo de la solución dada su complejidad y alcance. Por las ventajas que brinda *NetBeans 7.2* a los desarrolladores y por su condición de *software* libre se selecciona como IDE para el desarrollo de la propuesta de solución.

Entre los lenguajes para la propuesta de solución seleccionados se encuentran en primer lugar el Lenguaje Unificado de Modelado (UML), del lado del servidor los lenguajes de programación PHP 5.3.8 debido a que es el lenguaje utilizado en *Moodle* y del lado del cliente *JavaScript*, XHTML 1.0 y CCS 2.0.

El análisis de los fundamentos teóricos expuestos en el capítulo y la selección de lenguajes y herramientas realizada constituyen el punto de partida para la construcción de un sistema que resuelva la problemática planteada.

2

CAPÍTULO

Introducción

El presente capítulo tiene como objetivo describir las características fundamentales de la funcionalidad “Recolector”, ofreciendo la propuesta de solución para el desarrollo de una funcionalidad capaz de gestionar toda la información no utilizada en la plataforma de teleformación *Moodle 2.3.x*. Además, se exponen de forma detallada las características a tener en cuenta durante la implementación de la funcionalidad. Por último se muestran los resultados de las fases de exploración y planificación mediante las Historias de usuario y las tarjetas CRC.

Diagnóstico inicial

Para el estudio llevado a cabo se hizo necesaria la realización de una entrevista a los administradores de la plataforma de teleformación *Moodle*. En específico a el Ingeniero en Ciencias Informáticas Carlos Acosta Montejo y los Técnicos en Ciencias Informáticas Levian Diaz Camejo y Zunami Blanco Rodríguez, para tener conocimiento acerca de los principales problemas que esta presentaba con respecto a la información que no se utiliza en la misma.

Como resultado de la entrevista realizada se destaca la importancia y necesidad existente en la creación de la funcionalidad “Recolector”. Se arrojó como resultado que los administradores no tienen forma de calcular el tiempo en que los usuarios y las actividades o recursos están inactivos. Tampoco pueden determinar de forma rápida qué acciones tomar con los mismos, pues no existe la posibilidad de comunicarle a los usuarios y a los editores de los cursos que sus actividades y recursos no están siendo utilizados antes de optar por eliminarlos para liberar espacio en los servidores donde se aloja la aplicación. (Ver Anexo 2)

2.1 Descripción de los procesos vinculados al campo de acción

La descripción de los procesos vinculados al campo de acción permite una mayor comprensión del problema y la determinación adecuada de las necesidades del cliente. Por otro lado facilita el diseño de las funcionalidades necesarias para optimizar dichos procesos. A continuación se describen los procesos relacionados con la gestión de la información no utilizada en el EVA de manera que se aportan elementos para determinar las características deseadas para la funcionalidad “Recolector” para la versión 2.3.x de *Moodle*.

Gestión de la información no utilizada en la plataforma *Moodle* en su versión 2.3.x:

Este proceso tiene como objetivo determinar los usuarios, actividades y recursos inactivos, los cuales son identificados por los administradores de forma muy complicada. Seguidamente se detalla el flujo actual del proceso.

Usuarios inactivos: En el caso de los usuarios inactivos se debe recurrir a un listado donde se muestra de cada uno el nombre, el identificador, la dirección de correo, la ciudad, el país y la fecha del último acceso. A partir de estos datos (fundamentalmente la fecha del último acceso) el administrador debe calcular el tiempo de inactividad de cada usuario y decidir qué acción tomar. No es posible enviar una notificación al usuario, solo se permite eliminarlo de la plataforma.

Actividades o recursos no utilizados: Para determinar si una actividad o recurso de un curso no está siendo utilizado se procede de manera similar al proceso realizado con los usuarios inactivos. Se accede a la lista donde se encuentran todas las actividades y recursos por cada curso existente en la plataforma. Se determina la fecha del último acceso y se calcula el tiempo de inactividad. De igual manera el administrador solo puede optar por eliminar la actividad o recurso.

2.1.1 Propuesta del sistema

Para solucionar el problema planteado en la investigación se ha decidido implementar una funcionalidad que permita la gestión de la información no utilizada de los usuarios y cursos de la plataforma de teleformación *Moodle* en su versión 2.3.x. Una vez concluida la misma los administradores del EVA contarán con una herramienta que de gran ayuda para la obtención de reportes como por ejemplo: la cantidad de usuarios que hace un determinado tiempo no acceden a la plataforma o las actividades y los recursos inactivos

por cada curso. Además, esta funcionalidad permitirá enviar una notificación de correo a los usuarios inactivos antes de eliminarlos. También posibilitará el envío de una notificación de correo a los profesores editores de los cursos que posean actividades o recursos no utilizados, así como su eliminación.

En los epígrafes siguientes se muestran los resultados de las dos primeras fases definidas por la metodología de desarrollo XP: Exploración y Planificación y los artefactos generados durante las mismas.

2.2 Principales particularidades de la funcionalidad “Recolector”

2.2.1 Funcionalidades del “Recolector”

Una funcionalidad es una especificación de qué se debe implementar. Es una descripción de cómo se debe comportar el sistema, de un atributo o una propiedad. Una condición o capacidad que debe cumplir o poseer un sistema o componente de un sistema. (Rodríguez, 2012)

A continuación se describen las funcionalidades con las que contará el “Recolector”:

Funcionalidades (F) (Ver Anexo3)

F1: Determinar tiempo de inactividad límite.

F2: Listar usuarios pasivos.

F3: Enviar notificación a usuarios pasivos.

F4: Eliminar usuario.

F5: Mostrar actividades y recursos no utilizados.

F6: Enviar notificación a profesores editores de un curso.

F7: Eliminar actividades y recursos no utilizados.

2.2.2 Características del “Recolector”

Las características del sistema o componente definen aspectos que sin ser funcionalidades resultan deseables desde el punto de vista del usuario. Describen aspectos del sistema que son visibles por el usuario pero que no establecen una relación directa con el comportamiento funcional del mismo. A continuación se muestran las características que debe poseer la funcionalidad “Recolector”.

Características

- **Apariencia o interfaz externa:** La interfaz debe ser agradable e intuitiva para conseguir la satisfacción de los usuarios con la utilización de la funcionalidad.
- **Rendimiento:** La funcionalidad propuesta debe ser rápida y el tiempo de respuesta el mínimo posible.
- **Mantenibilidad:** Utilizar estándares para el desarrollo de aplicaciones Web (XHTML 1.0 y CSS 2.0).
- **Seguridad:**
 - El acceso al sistema se restringirá a los usuarios de acuerdo al rol que desempeñen.
 - La funcionalidad “Recolector” utilizará el protocolo ADO para establecer la conexión con el sistema de base de datos.
- **Software:** La funcionalidad se elaborará para la versión 2.3.x de *Moodle*. El sistema se implementará con tecnología PHP 5.3.8 utilizando como entorno de desarrollo al IDE *NetBeans 7.2*.

2.2.3 Personal relacionado con la funcionalidad “Recolector”

El personal relacionado con una aplicación son todas aquellas personas que intervienen en el desarrollo de la misma o harán uso de esta. La funcionalidad “Recolector” ha sido especialmente diseñada para responder a una necesidad de los administradores del EVA, que son por ende las personas vinculadas a la misma.

Personal Relacionado	Justificación
Administrador	Es aquella persona que hará uso de la funcionalidad para lograr una correcta gestión de la información no utilizada en el EVA (personalización de <i>Moodle 2.3.x</i>) de la Universidad de las Ciencias Informáticas.

Tabla 1: Descripción del personal relacionado con la aplicación.

2.3 Fase Exploración

Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas “Historias de usuario”. Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (pues estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalle en cada iteración.

Exploración y Planificación

Esta fase dura típicamente un par de semanas, y el resultado es una visión general del sistema, y un plazo total estimado. (Joskowicz, 2008)

2.3.1 Historias de usuario

La definición del Modelo de Historia de usuario (HU) del Negocio, es una de las actividades más importantes dentro de la metodología XP, pues en él se definen las características específicas del negocio en cuestión. Seguidamente se encuentran las Historias de usuario las cuales responden a cada funcionalidad del sistema.

Historia de usuario	
Código: Número sucesivo a partir del uno de la HU.	Nombre de la Historia de usuario: Identifica la HU.
Modificación de la HU Número: Identifica si la HU es una modificación de otra HU.	
Programador: El usuario del sistema que utiliza o protagoniza la HU.	Iteración Asignada: Precisa la iteración a la que pertenece la HU.
Prioridad: Define la relevancia e impacto de la HU para el negocio de acuerdo a las necesidades del usuario.	Puntos Estimados: Permiten estimar la duración de implementación.
Riesgo en Desarrollo: Define la dificultad técnica que supone desarrollar la HU desde el punto de vista del programador.	Puntos Reales: Define el tiempo real de duración de la implementación de esta funcionalidad.
Descripción: Explica en qué consiste la HU, teniendo en cuenta las acciones realizadas por el usuario y la respuesta brindada por el sistema.	
Observaciones: Información extra que se estime agregar para hacer más comprensible la Historia de usuario. Por ejemplo: conceptos, pos condiciones, entre otros.	

Tabla 2: Estructura de una Historia de usuario.

Historia de usuario	
Código: HU1	Nombre Historia de usuario: Determinar tiempo de inactividad límite
Modificación de Historia de usuario Número: Ninguna	
Referencia: Ítem FS1	
Programador: Liuvís Moreno Díaz	Iteración Asignada: Primera
Prioridad: Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Bajo	Puntos Reales: 1 semana
Descripción: La Historia de usuario permite determinar el tiempo límite en que los usuarios, las actividades y	

recursos pueden estar inactivos. Para realizar esta acción el administrador accede al bloque de administración, específicamente a la opción “Recolector”. Seguidamente el sistema muestra una página donde el administrador podrá seleccionar el tiempo de inactividad límite de cada usuario, actividad o recurso de cada curso (este tiempo podrá ser 6 meses, 1 año o 2 años, debido a que fue el tiempo recomendado por el cliente, pues por lo general el curso está estructurado por un año que son dos semestres de 6 meses).

Observaciones:

Tabla 3: Descripción de la HU, determinar tiempo de inactividad límite.

Historia de usuario	
Código: HU2	Nombre Historia de usuario: Listar usuarios pasivos
Modificación de Historia de usuario Número: Ninguna	
Referencia: Ítem FS2	
Programador: Liuvís Moreno Díaz	Iteración Asignada: Primera
Prioridad: Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Bajo	Puntos Reales: 1 semana
Descripción: La Historia de usuario permite listar todos los usuarios pasivos existentes en la plataforma. Para realizar esta acción el administrador accede al bloque de administración y selecciona la funcionalidad “Recolector”. Seguidamente el sistema muestra una página con varias opciones (“Listar usuarios pasivos” y “Listar actividades y recursos no utilizados”). El administrador elige la opción “Listar usuarios pasivos”. El sistema muestra una página con una tabla en la cual se encuentran todos los usuarios que desde hace un tiempo determinado no acceden a la plataforma (este tiempo lo define el administrador del sitio, ver HU1).	
Observaciones: Si no existe ningún usuario inactivo, el sistema muestra una interfaz donde aparecerá un cartel “No existe ningún usuario inactivo en la plataforma”.	

Tabla 4: Descripción de la HU, listar usuarios pasivos.

Historia de usuario	
Código: HU3	Nombre Historia de usuario: Enviar notificación a usuarios pasivos
Modificación de Historia de usuario Número: Ninguna	
Referencia: Ítem FS3	
Programador: Liuvís Moreno Díaz	Iteración Asignada: Segunda
Prioridad: Media	Puntos Estimados: 1 semanas
Riesgo en Desarrollo: Bajo	Puntos Reales: 1 semanas
Descripción: La Historia de usuario tiene como objetivo permitir enviar una notificación de correo al usuario,	

poniéndolo al tanto de que hace algún tiempo no accede a la plataforma. Para realizar esta acción el administrador accede al bloque de administración y se selecciona la funcionalidad “Recolector”. Seguidamente el sistema muestra una página con varias opciones (“Listar usuarios pasivos” y “Listar actividades y recursos no utilizados”). El administrador elige a la opción “Listar usuarios pasivos” y el sistema muestra una página con una tabla en la cual se encuentran todos los usuarios que desde hace un tiempo determinado no acceden a la plataforma (este tiempo lo define el administrador del sitio, ver HU1), debajo de la tabla se encuentran dos botones “Seleccionar todo” y “No seleccionar todo”. El administrador selecciona los usuarios a los que decide enviarles la notificación de correo y seguidamente la opción “Enviar notificación”. El sistema muestra una página donde se podrá redactar el correo que será enviado al usuario, la cual posee un botón “Vista Previa” al cual el administrador da clic y se muestra una vista previa del correo. Luego el administrador da clic en la opción “Enviar mensaje” si decide enviarlo o “Cancelar” si no lo desea. Por último el sistema envía el correo electrónico.

Observaciones: Si no existe ningún usuario inactivo, el sistema muestra una interfaz donde aparecerá un cartel “No existe ningún usuario inactivo en la plataforma”.
Si el cuerpo del correo está vacío se muestra un mensaje “Todos los campos son obligatorios”.

Tabla 5: Descripción de la HU, enviar notificación.

Historia de usuario	
Código: HU4	Nombre Historia de usuario: Eliminar usuarios
Modificación de Historia de usuario Número: Ninguna	
Referencia: Ítem FS4	
Programador: Liuvís Moreno Díaz	Iteración Asignada: Tercera
Prioridad: Media	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Bajo	Puntos Reales: 1 semana
Descripción:	
<p>La Historia de usuario permite eliminar todos los usuarios pasivos existentes en la plataforma. Para realizar esta acción el administrador accede al bloque de administración y selecciona la funcionalidad llamada “Recolector”. Seguidamente el sistema muestra una página con varias opciones (“Listar usuarios pasivos” y “Listar actividades y recursos no utilizados”). El administrador elige la opción “Listar usuarios pasivos” y el sistema muestra una página con una tabla en la cual se encuentran todos los usuarios que desde hace un tiempo determinado no acceden a la plataforma (este tiempo lo define el administrador del sitio, ver HU1), debajo de la tabla se encuentran dos botones “Seleccionar todo” y “No seleccionar todo”, el administrador selecciona los usuarios a los que decide eliminar de la plataforma y seguidamente la opción “Borrar”. El sistema muestra una página donde aparecen las opciones “Continuar” si decide aún eliminar a los usuarios o “Cancelar” si no lo desea. El administrador da clic en “Continuar” y se eliminan por completo los usuarios de la plataforma.</p>	

Observaciones: Si no existe ningún usuario inactivo, el sistema muestra una interfaz donde aparecerá un cartel “No existe ningún usuario inactivo en la plataforma”.

Tabla 6: Descripción de la HU, eliminar usuarios.

Historia de usuario	
Código: HU5	Nombre Historia de usuario: Mostrar actividades y recursos no utilizados
Modificación de Historia de usuario Número: Ninguna	
Referencia: Ítem FS5	
Programador: Liuvís Moreno Díaz	Iteración Asignada: Primera
Prioridad: Alta	Puntos Estimados: 2 semanas
Riesgo en Desarrollo: Bajo	Puntos Reales: 2 semanas
Descripción: La Historia de usuario debe permitir mostrar un listado con todas las actividades y recursos pasivos existentes en la plataforma. Para realizar esta acción el administrador accede al bloque de administración y selecciona la funcionalidad “Recolector”. Seguidamente se muestra una página con varias opciones (“Listar usuarios pasivos” y “Listar actividades y recursos no utilizados”). El administrador elige la opción “Listar actividades y recursos no utilizados”. El sistema muestra una página con una tabla en la cual se encuentran por cada curso las actividades y recursos que no están siendo utilizados en la plataforma.	
Observaciones: Si no existe ninguna actividad o recurso inactivo, el sistema muestra una interfaz donde aparecerá un cartel “No existe ninguna actividad o recurso inactivo en la plataforma”.	

Tabla 7: Descripción de la HU, mostrar actividades y recursos no utilizados.

Historia de usuario	
Código: HU6	Nombre Historia de usuario: Enviar notificación a profesores editores de un curso
Modificación de Historia de usuario Número: Ninguna	
Referencia: Ítem FS6	
Programador: Liuvís Moreno Díaz	Iteración Asignada: Segunda
Prioridad: Media	Puntos Estimados: 2 semanas
Riesgo en Desarrollo: Bajo	Puntos Reales: 2 semana
Descripción: La Historia de usuario tiene como objetivo permitir enviar un correo electrónico a los profesores editores del curso, poniéndolo al tanto de las actividades o los recursos que no están siendo utilizados en la plataforma. Para realizar esta acción el administrador accede al bloque de administración y selecciona la funcionalidad llamada “Recolector”. Seguidamente el sistema muestra una página con varias opciones (“Listar usuarios pasivos” y “Listar actividades y recursos no	

utilizados”). El administrador elige a la opción “Listar actividades y recursos no utilizados”, (el tiempo en que las actividades o recursos están inactivos lo define el administrador del sitio, ver HU1). El sistema muestra una página con una tabla en la cual se encuentran todas las actividades y recursos que no están siendo utilizados en la plataforma por cada curso. En tabla también se encuentra al lado de cada curso una opción llamada “Notificar profesor” a la cual el administrador da clic. El sistema muestra una página donde se podrá redactar el correo que será enviado al profesor editor, la misma posee un botón “Vista Previa” al cual el administrador da clic y se muestra una vista previa del correo. Luego el administrador da clic en la opción “Enviar mensaje” si decide enviarlo o “Cancelar” si no lo desea. Por último el sistema envía el correo electrónico.

Observaciones: Si no existe ninguna actividad o recurso inactivo, el sistema muestra una interfaz donde aparecerá un cartel “No existe ninguna actividad o recurso no utilizado en la plataforma”. Si el cuerpo del correo está vacío se muestra un mensaje “Todos los campos son obligatorios”.

Tabla 8: Descripción de la HU, enviar notificación a profesores editores de un curso.

Historia de usuario	
Código: HU7	Nombre Historia de usuario: Eliminar actividades y recursos no utilizados
Modificación de Historia de usuario Número: Ninguna	
Referencia: Ítem FS7	
Programador: Liuvís Moreno Díaz	Iteración Asignada: Tercera
Prioridad: Media	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Bajo	Puntos Reales: 1 semana
<p>Descripción:</p> <p>La Historia de usuario tiene como objetivo permitir eliminar las actividades o recursos que no están siendo utilizados en la plataforma. Para realizar esta acción el administrador accede al bloque de administración y selecciona la funcionalidad llamada “Recolector”. Seguidamente el sistema muestra una página con varias opciones (“Listar usuarios pasivos” y “Listar actividades y recursos no utilizados”). El administrador elige a la opción “Listar actividades y recursos no utilizados”, (el tiempo en que las actividades o recursos están inactivos lo define el administrador del sitio, ver HU1). El sistema muestra una página con una tabla en la cual se encuentran todas las actividades y recursos que no están siendo utilizados en la plataforma por cada curso. Debajo de la tabla se encuentran dos botones “Seleccionar todo” y “No seleccionar todo”, el administrador selecciona las actividades y recursos que decide eliminar de la plataforma y seguidamente la opción “Borrar”. El sistema muestra una página donde aparecen las opciones “Continuar” si decide aun eliminar las actividades y recursos o “Cancelar” si no lo desea. El administrador da clic en “Continuar” y se eliminan por completo las actividades y recursos no utilizados en la plataforma.</p>	

Observaciones: Si no existe ninguna actividad o recurso inactivo, el sistema muestra una interfaz donde aparecerá un cartel “No existe ninguna actividad o recurso inactivo en la plataforma”.

Tabla 9: Descripción de la HU, eliminar actividades y recursos no utilizados.

Para comprender la duración de las estimaciones de las Historias de usuario anteriores, es necesario aclarar que 1 semana laboral equivale a 5 días laborales. Se hace necesario realizar esta aclaración pues en ocasiones se cometen errores de cálculo al estimar los tiempos de desarrollo en base a los 7 días de la semana.

2.3.2 Estimación

Las estimaciones del esfuerzo para implementar las Historias de usuario permiten tener una medida bastante real de la velocidad de progreso del proyecto y brindan una guía razonable a la cual ajustarse. En la tabla siguiente se muestra el esfuerzo estimado por Historia de usuario para el desarrollo de la funcionalidad “Recolector”, expresado en semanas.

2.3.3 Estimación de esfuerzos por Historias de usuario

Historia de usuario	Estimación (semanas)
Determinar el tiempo de inactividad límite	1
Listar usuarios pasivos	1
Enviar notificación a usuarios pasivos	1
Eliminar usuarios	1
Mostrar actividades y recursos no utilizados	2
Enviar notificación a profesores editores de un curso	2
Eliminar actividades y recursos no utilizados	1

Tabla 10: Estimación de esfuerzos por Historias de usuario.

2.4 Fase de planificación

La planificación es una fase corta, en la que se determina el orden en que deberán implementarse las Historias de usuario y asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado fundamental de esta fase es un Plan de Entregas, o “*Release Plan*”. (Joskowicz, 2008)

A partir de una reunión de planificación con los administradores del EVA, se estableció un plan o cronograma de entregas, para luego comenzar la fase de iteraciones. Los epígrafes siguientes exponen los resultados y artefactos generados durante la fase de

planificación. Primeramente se muestra el Plan de iteraciones, en segundo lugar el Plan de duración de las iteraciones y finalmente el Plan de entregas.

2.4.1 Plan de iteraciones

En este plan las Historias de usuario seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada Historia de usuario se traduce en tareas específicas de programación. Asimismo, para cada Historia de usuario se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores. (Joskowicz, 2008)

Una iteración no es más que un mini-proyecto en el que se obtiene un resultado en *software* con un valor para el cliente, en este caso los administradores del EVA. Resulta válido aclarar que al finalizar una iteración no se obtiene una versión total del producto, pues este se completa finalizar la última iteración.

La implementación de la funcionalidad “Recolector” fue planificada en tres iteraciones, las cuales se detallan a continuación:

Iteración 1

Esta iteración tiene como objetivo la implementación de las Historias de usuario con mayor prioridad. Al finalizar se contará con las funcionalidades descritas en las Historias de usuario 1, 2 y 5 las cuales hacen alusión a determinar tiempo de inactividad límite, listar usuarios pasivos y mostrar actividades y recursos no utilizados respectivamente. Además, se tendrá la primera versión de prueba, la cual será mostrada al cliente con el objetivo de obtener una retroalimentación para el grupo de trabajo.

Iteración 2

El objetivo de esta iteración es la implementación de las funcionalidades con prioridad media. Con la culminación de la misma se tendrán implementadas las peticiones del cliente descritas en las Historias de usuario 3 y 6, en las cuales se hace mención a enviar notificación a usuarios pasivos y enviar notificación a profesores editores de un curso respectivamente.

Iteración 3

El objetivo de esta iteración es la implementación de las funcionalidades con prioridad media restantes. Con la culminación de la misma se tendrán implementadas las peticiones del cliente descritas en las Historias de usuario 4 y 7, en las cuales se hace mención a eliminar usuarios y eliminar actividades o recursos no utilizados respectivamente.

2.4.2 Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto utilizando XP se crea el plan de duración de cada una de las iteraciones. En este plan se especifica más detalladamente el orden de desarrollo de las Historias de usuario dentro de cada iteración así como la estimación completa de la misma.

Iteración	Historias de usuario	Tiempo (semanas)
1	<ol style="list-style-type: none">1. Determinar el tiempo de inactividad límite.2. Listar usuarios pasivos.3. Mostrar actividades y recursos no utilizados.	4
2	<ol style="list-style-type: none">1. Enviar notificación a usuarios pasivos.2. Enviar notificación a profesores editores de un curso.	3
3	<ol style="list-style-type: none">1. Eliminar usuarios.2. Eliminar actividades y recursos no utilizados.	2

Tabla 11: Distribución de Historias de usuario por iteración.

2.4.3 Plan de entregas

El cronograma de entregas establece qué Historias de usuario serán agrupadas para conformar una entrega y el orden de las mismas. (Joskowicz, 2008)

Este cronograma es el resultado de una reunión (Reunión de planeamiento) entre todos los actores del proyecto, en este caso la desarrolladora y los clientes, o sea, los administradores del EVA.

Entregable	Final 1ra iteración	Final 2da iteración	Final 3ra iteración
------------	---------------------	---------------------	---------------------

Funcionalidad "Recolector"	01/03/2013	29/03/2013	26/04/2013
-------------------------------	------------	------------	------------

Tabla 12: Plan de Entrega de la funcionalidad "Recolector".

2.4.4 Diseño del sistema

La metodología seleccionada no requiere la descripción del sistema por medio de diagramas de clases utilizando notación UML, esta se guía por técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). Esto no implica que no se utilicen los diagramas para obtener una mejor visión y comunicación entre el equipo de trabajo, siempre y cuando su complejidad no sea alta y defina información importante.

En los epígrafes que siguen se exponen los artefactos generados durante el diseño de la funcionalidad "Recolector". En este caso las tarjetas CRC y el diagrama de despliegue. También se abordan las características de la arquitectura utilizada.

Tarjetas CRC

Las características más sobresalientes de las tarjetas CRC son su simpleza y ductilidad. Una tarjeta CRC no es más que una ficha de papel o cartón que representa a una entidad del sistema. (Casas, y otros, 2008)

Estas tarjetas se utilizan para estructurar las clases y a su vez definir las responsabilidades sobre las mismas, así como la simulación de escenarios en el sistema.

Tarjeta CRC	
Clase: Nombre de la clase que se está modelando.	
Responsabilidades: Es una descripción de alto nivel del propósito de la clase.	Colaboración: Indica con qué otras clases se requieren relación para cumplir la responsabilidad.

Tabla 13: Estructura de las tarjetas CRC.

Seguidamente se muestran las tarjetas CRC que describen las clases utilizadas para la implementación de la funcionalidad "Recolector".

Tarjeta CRC	
Clase: users_listing	
Responsabilidades: Es la encargada de listar los	Colaboración:

Exploración y Planificación

usuarios inactivos que se encuentran dentro de la plataforma.	<ul style="list-style-type: none">• user_filtering• sql_filter
---	---

Tabla 14: Descripción de la tarjeta CRC users_listing.

Tarjeta CRC	
Clase: message_post_message	
Responsabilidades: Es la encargada de enviar notificación de correo a los usuarios inactivos dentro de la plataforma.	Colaboración: <ul style="list-style-type: none">• emailto• emailselect• data_submitted• preg_match

Tabla 15: Descripción de la tarjeta CRC message_post_message.

Tarjeta CRC	
Clase: delete_user	
Responsabilidades: Es la encargada de eliminar los usuarios inactivos que se encuentran dentro de la plataforma.	Colaboración: <ul style="list-style-type: none">• confirm• data_submitted• preg_match

Tabla 16: Descripción de la tarjeta CRC delete_user.

Tarjeta CRC	
Clase: courses	
Responsabilidades: Es la encargada de listar todos los cursos inactivos que se encuentran dentro de la plataforma.	Colaboración:

Tabla 17: Descripción de la tarjeta CRC courses.

Tarjeta CRC	
Clase: fast_modinfo	
Responsabilidades: Es la encargada de devolver la información de los módulos de un curso, el cual es pasado por parámetro.	Colaboración:

Tabla 18: Descripción de la tarjeta CRC fast_modinfo.

Tarjeta CRC	
Clase: delete_course_module	
Responsabilidades: Es la encargada de eliminar	Colaboración:

las actividades y recursos inactivos que se encuentran dentro de la plataforma.	<ul style="list-style-type: none">• confirm• data_submitted• preg_match
---	---

Tabla 19: Descripción de la tarjeta CRC delete_course_module.

Arquitectura del sistema

La arquitectura de software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

En XP, el objetivo es guiar todos los desarrollos compartiendo una historia simple de cómo trabaja el sistema. Comúnmente el sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Este lenguaje de metáforas se utiliza para describir la arquitectura del sistema, ayudando a la comunicación entre el personal involucrado en el proyecto. (Fowler, y otros, 2010)

La metodología XP propone además que los modelos sean algo tan simple como un dibujo o bosquejo que se encuentren a disposición de todos los miembros del equipo, visible para el público, porque a pesar de que las metáforas pueden ser muy eficaces, un modelo arquitectónico a menudo proporciona el mayor detalle que el equipo necesita. (Gou, y otros, 2010)

De estos criterios se aplicó el segundo pues resultó muy útil y entendible para la desarrolladora y el cliente la representación de la arquitectura empleada a través del esquema mostrado en la Figura 1.

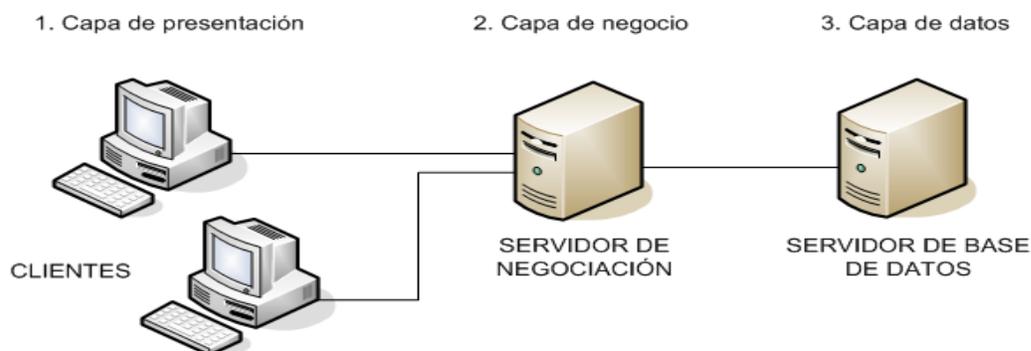


Figura 1: Esquema de la Arquitectura 3 capas.

Existen diferentes tipos de arquitecturas para los sistemas a nivel mundial, pero la utilizada por la plataforma de teleformación Moodle es la arquitectura de tres niveles, también conocida como arquitectura de tres capas. Esta define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa solo pueden hacer referencia a componentes en capas inmediatamente inferiores. Esta posee 3 capas:

1. **Capa de presentación:** Es la que ve el usuario y le presenta el sistema. Le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. En esta se encuentra la Interfaz Web de la publicación, que es el medio de comunicación directo entre el sistema y el usuario. Esta capa se comunica únicamente con la capa de negocio.
2. **Capa de negocio:** Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta envuelve el software de aplicación usado en la programación de la solución. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.
3. **Capa de acceso a datos:** Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. (Calle, 2011)

Conclusiones Parciales

A partir de la descripción de los procesos vinculados al campo de acción se elaboró una propuesta de solución de las necesidades de los administradores del EVA.

La definición y descripción de las características que debe poseer la funcionalidad "Recolector" de conjunto con las Historias de usuario constituyen entregables vitales para guiar la verdadera fase de implementación.

Exploración y Planificación

La elaboración del Plan de iteraciones y Plan de entregas resultan artefactos claves para la organización del trabajo, ya que ordenarán qué implementar en cada momento.

La determinación del esfuerzo necesario para concluir el desarrollo arrojó un tiempo de 9 semanas para el cumplimiento de las tareas planteadas y por consiguiente la implementación de la funcionalidad “Recolector”.

En las tarjetas CRC se han descrito las clases utilizadas y sus colaboradoras de modo que se facilitará la implementación de las mismas.

La funcionalidad se implementará haciendo uso de una arquitectura de tres capas, la cual deriva en buenas prácticas de programación y está acorde a las características de la plataforma *Moodle*.

Introducción

El presente capítulo tiene como objetivo describir la fase implementación de la funcionalidad “Recolector”. Se exponen los resultados de las 3 iteraciones realizadas durante las cuales se obtuvo un producto que fue mostrado al cliente en cada caso. Además, se muestran los casos de prueba basados en historias de usuario y los resultados obtenidos una vez realizadas las pruebas.

3.1 Fase de implementación

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas generando al final de cada iteración un entregable funcional que implementa las Historias de usuario asignadas a la misma. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance. (Joskowicz, 2008)

Para el desarrollo del “Recolector” se realizaron tres iteraciones, obteniéndose finalmente una funcionalidad con las características deseadas, según la definición realizada. A continuación se detallan cada una de las iteraciones.

3.1.1 Iteración 1

En esta iteración se implementan las Historias de usuario 1, 2 y 5 seguidamente se muestran las tareas por Historia de usuario definidas en la primera iteración.

Tarea de Ingeniería	
Número de tarea: 1	Número de HU: 1
Nombre de la tarea: Determinar tiempo límite de inactividad	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha de inicio: 14/01/2013	Fecha final: 18/01/2013
Programador responsable: Liuvís Moreno Díaz	

Implementación y Prueba

Descripción: El administrador determina el tiempo límite en que el usuario y las actividades o recursos de los cursos están inactivos (este tiempo podrá ser 6 meses, 1 año o 2 años, debido a que fue el tiempo recomendado por el cliente, pues por lo general el curso está estructurado por un año que son dos semestres de 6 meses).

Tabla 20: Tarea, determinar tiempo límite de inactividad.

Tarea de Ingeniería	
Número de tarea: 2	Número de HU: 2
Nombre de la tarea: Listar usuarios pasivos	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha de inicio: 21/01/2013	Fecha final: 25/01/2013
Programador responsable: Liuvis Moreno Diaz	
Descripción: El administrador podrá listar todos los usuarios pasivos o inactivos existentes en la plataforma dependiendo del tiempo de inactividad de los mismos.	

Tabla 21: Tarea, listar usuarios pasivos.

Tarea de Ingeniería	
Número de tarea: 3	Número de HU: 5
Nombre de la tarea: Mostrar actividades y recursos no utilizados	
Tipo de tarea: Desarrollo	Puntos de estimación: 2
Fecha de inicio: 04/02/2013	Fecha final: 15/02 /2013
Programador responsable: Liuvis Moreno Diaz	
Descripción: El administrador lista por cada curso todas las actividades y recursos pasivos existentes en la plataforma dependiendo del tiempo de inactividad de los mismos.	

Tabla 22: Tarea, mostrar actividades y recursos no utilizados.

3.1.2 Iteración 2

En esta iteración se implementan las Historias de usuario 3 y 6. Seguidamente se muestran las tareas por Historia de usuario definidas en la segunda iteración.

Tarea de Ingeniería	
Número de tarea: 4	Número de HU: 3
Nombre de la tarea: Enviar notificación a usuarios pasivos	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha de inicio: 25/03/2013	Fecha final: 29/03/2013
Programador responsable: Liuvis Moreno Diaz	

Implementación y Prueba

Descripción: El administrador envía una notificación de correo a todos los usuarios pasivos existentes en la plataforma informándolos del tiempo transcurrido en que no acceden a la misma.

Tabla 23: Tarea, enviar notificación a usuarios pasivos.

Tarea de Ingeniería	
Número de tarea: 5	Número de HU: 6
Nombre de la tarea: Enviar notificación a profesores editores de un curso	
Tipo de tarea: Desarrollo	Puntos de estimación: 2
Fecha de inicio: 01/04/2013	Fecha final: 05/04/2013
Programador responsable: Liuvis Moreno Diaz	
Descripción: El administrador envía una notificación de correo a todos los profesores editores del curso informándolos del tiempo transcurrido sin que los usuarios accedan a las actividad o recurso de los cursos.	

Tabla 24: Tarea, enviar notificación a profesores editores de un curso.

3.1.3 Iteración 3

En esta iteración se implementan las Historias de usuario 4 y 7. Seguidamente se muestran las tareas por Historia de usuario definidas en la tercera iteración.

Tarea de Ingeniería	
Número de tarea: 6	Número de HU: 4
Nombre de la tarea: Eliminar usuario	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha de inicio: 08/04/2013	Fecha final: 12/04/2013
Programador responsable: Liuvis Moreno Diaz	
Descripción: El administrador elimina a todos los usuarios inactivos existentes en la plataforma.	
Observación: Si no existe ningún usuario inactivo, el sistema muestra una interfaz donde aparecerá un cartel "No existe ningún usuario inactivo en la plataforma".	

Tabla 25: Tarea, eliminar usuario.

Tarea de Ingeniería	
Número de tarea: 7	Número de HU: 7
Nombre de la tarea: Eliminar actividades y recursos no utilizados	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha de inicio: 22/04/2013	Fecha final: 26/04/2013
Programador responsable: Liuvis Moreno Diaz	

Descripción: El administrador elimina todas las actividades y recursos no utilizados dentro de la plataforma.

Observación: Si no existe ninguna actividad o recurso que no estén siendo utilizados, el sistema muestra una interfaz donde aparecerá un cartel "No existen actividades o recursos sin utilizar en la plataforma".

Tabla 26: Tarea, eliminar actividades y recursos inactivos.

3.1.4 Diagrama de despliegue

Para un mejor entendimiento de las tareas, flujos y métodos de desarrollo de las funcionalidades, se pueden crear diagramas, siempre que su creación no implique mayor esfuerzo que la implementación del mismo. Siguiendo este principio, se elaboró el diagrama de despliegue que permite apreciar de forma visual cómo se encuentran relacionados físicamente los componentes en la aplicación.

Un diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema, en términos de cómo se distribuye la funcionalidad entre nodos de cómputo. (Jacobson, y otros, 2000)

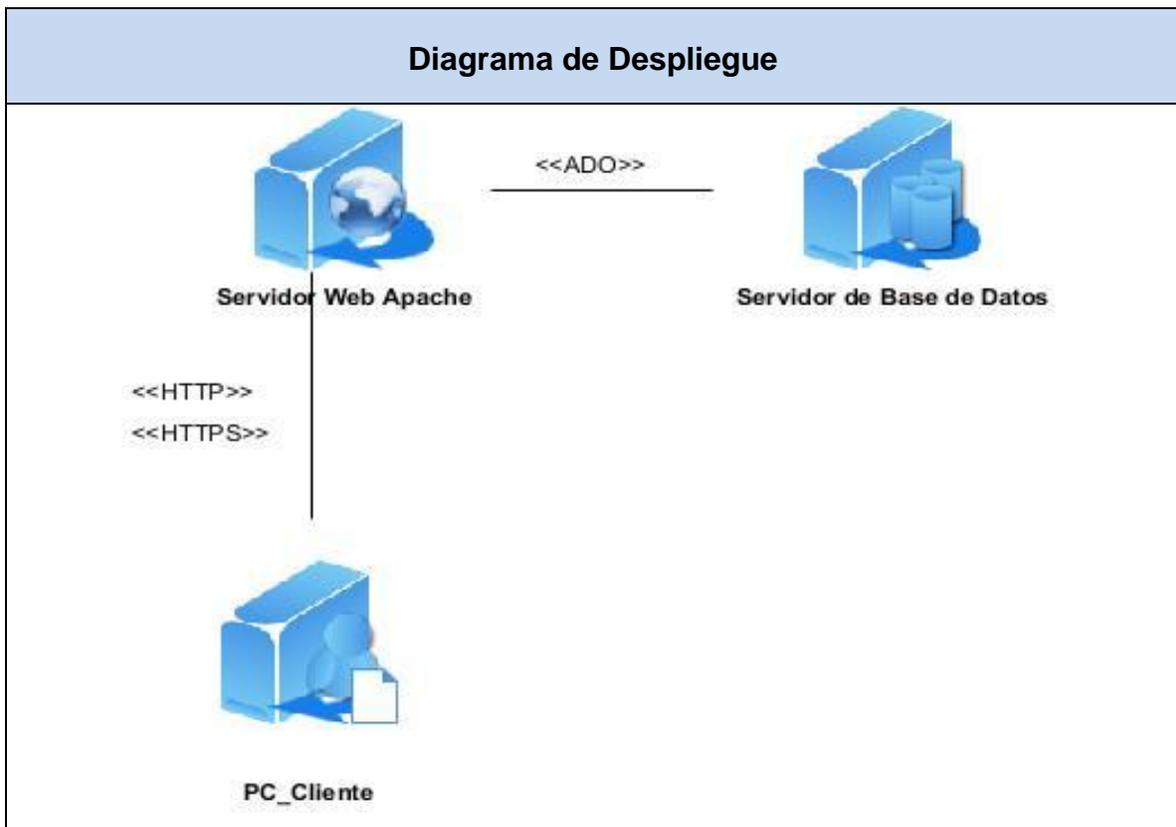


Figura 2: Diagrama de despliegue.

3.2 Fase de Prueba

Uno de los pilares de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos implementados. La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente. (Allende, 2006)

Pruebas unitarias

Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo. (Joskowicz, 2008)

Pruebas de aceptación

Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una Historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. (Joskowicz, 2008)

Se considera que en el caso del desarrollo de la funcionalidad “Recolector” las pruebas de aceptación poseen un peso superior a las unitarias. El “Recolector” es el resultado de la implementación realizada por una persona por lo que no es necesario velar por la propiedad colectiva del código. Por otro lado las pruebas de aceptación arrojan a vista de los desarrolladores la satisfacción del cliente. Sobre el fundamento de estas razones se decidió realizar pruebas de aceptación para comprobar la factibilidad de las funcionalidades implementadas.

3.2.1 Pruebas de aceptación

Durante las iteraciones, las Historias de usuario correspondientes se expresan en pruebas de aceptación. El cliente desde su punto de vista creará los casos de prueba en

Implementación y Prueba

conjunto con los desarrolladores para probar que una HU ha sido implementada correctamente. Por cada HU se elaboran todas las pruebas de aceptación que se necesiten para asegurar su correcto funcionamiento. El objetivo final de estas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable. (Beck, 2012)

A continuación se muestran los casos de prueba de aceptación basados en Historias de usuario para la funcionalidad “Recolector”. La siguiente descripción fue elaborada por la desarrolladora de conjunto con el cliente (administradores del EVA) y llevados a cabo por el cliente.

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de usuario: 1
Nombre Historia de usuario: Determinar el tiempo de inactividad límite	
Descripción: Prueba para la funcionalidad determinar el tiempo límite en que el usuario y las actividades o recursos puede estar inactivos.	
Acciones del Usuario	Respuesta del Sistema
1. El administrador accede al bloque de administración y selecciona la funcionalidad “Recolector”	1.1 El sistema muestra una página donde aparecerá un filtro para seleccionar el tiempo de inactividad, el cual estará definido por (6 meses, 1 año o 2 años).
2. El administrador elige el tiempo seleccionando una de estas opciones (6 meses, 1 año o 2 años) para que los usuarios, actividades y recursos estén inactivos.	2.1 El sistema define los usuarios, actividades y recursos que estén en ese tiempo como inactivos.
Resultado esperado: El tiempo es determinado según las necesidades de los administradores sin mucha complejidad.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 27: Prueba de aceptación 1.

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de usuario: 2
Nombre Historia de usuario: Listar usuarios pasivos	
Descripción: Prueba para la funcionalidad, listar usuarios pasivos.	
Condiciones de ejecución: Debe existir al menos un usuario inactivo.	
Acciones del Usuario	Respuesta del Sistema

Implementación y Prueba

1. El administrador accede al bloque de administración y selecciona la funcionalidad "Recolector"	1.1 El sistema muestra una página con varias opciones ("Listar usuarios pasivos" y "Listar actividades y recursos no utilizados")
2. El administrador elige la opción "Listar usuarios pasivos"	2.1 El sistema muestra una página con una tabla en la cual se encuentran todos los usuarios que hace un tiempo determinado por el administrador no acceden a la plataforma.
Resultado esperado: La lista de usuarios inactivos es mostrada de forma correcta en dependencia del tiempo de inactividad determinado por el administrador de la plataforma.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 28: Prueba de aceptación 2.

Caso de Prueba de Aceptación	
Código: HU3_P1	Historia de usuario: 3
Nombre Historia de usuario: Enviar notificación a usuarios pasivos	
Descripción: Prueba para la funcionalidad, enviar notificación a usuarios pasivos.	
Condiciones de ejecución: Debe existir al menos un usuario inactivo.	
Acciones del Usuario	Respuesta del Sistema
1. El administrador accede al bloque de administración y selecciona la funcionalidad "Recolector"	1.1 El sistema muestra una página con varias opciones ("Listar usuarios pasivos" y "Listar actividades y recursos no utilizados")
2. El administrador elige la opción "Listar usuarios pasivos"	2.1 El sistema muestra una página con una tabla en la cual se encuentran todos los usuarios que hace un tiempo determinado por el administrador no acceden a la plataforma.
3. El administrador selecciona cualquiera de las siguientes acciones: a) Seleccionar el botón "Seleccionar todo". b) Seleccionar solo los usuarios a los cuales se desea enviar la notificación de correo.	
4. El administrador selecciona la opción "Enviar notificación".	4.1 El sistema muestra una página donde se podrá redactar el correo que será enviado al usuario.
5. El administrador da clic en el botón "Vista Previa".	5.1 El sistema muestra una vista previa del correo.
6. El administrador da clic en el botón "Enviar".	6.1 El sistema envía la notificación de correo.
Curso Alternativo	

Implementación y Prueba

5. El administrador da clic en el botón "Cancelar".	5.1 El sistema vuelve a la página principal.
Resultado esperado: El correo electrónico es enviado de forma correcta a los usuarios inactivos de la plataforma.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 29: Prueba de aceptación 3.

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de usuario: 4
Nombre Historia de usuario: Eliminar usuario	
Descripción: Prueba para la funcionalidad, eliminar usuario.	
Condiciones de ejecución: Debe existir al menos un usuario inactivo.	
Acciones del Usuario	Respuesta del Sistema
1. El administrador accede al bloque de administración y selecciona la funcionalidad "Recolector"	1.1 El sistema muestra una página con varias opciones ("Listar usuarios pasivos" y "Listar actividades y recursos no utilizados")
2. El administrador elige la opción "Listar usuarios pasivos"	2.1 El sistema muestra una página con una tabla en la cual se encuentran todos los usuarios que hace un tiempo determinado por el administrador no acceden a la plataforma.
3. El administrador selecciona cualquiera de las siguientes acciones: a) Seleccionar el botón "Seleccionar todo". b) Seleccionar solo los usuarios a los cuales se desea eliminar de la plataforma.	
4. El administrador selecciona la opción "Borrar".	4.1 El sistema muestra una página donde se encuentran dos botones "Continuar" y "Cancelar".
5. El administrador da clic en el botón "Continuar".	5.1 El sistema elimina los usuarios inactivos de la plataforma.
Curso Alternativo	
5. El administrador da clic en el botón "Cancelar".	5.1 El sistema vuelve a la página principal.
Resultado esperado: El usuario pasivo es eliminado por completo de la plataforma de manera correcta.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 30: Prueba de aceptación 4.

Caso de Prueba de Aceptación

Implementación y Prueba

Código: HU5_P1	Historia de usuario: 5
Nombre Historia de usuario: Mostrar actividades y recursos no utilizados	
Descripción: Prueba para la funcionalidad, Mostrar actividades y recursos no utilizados.	
Condiciones de ejecución: Debe existir al menos una actividad o un recurso inactivo de un curso determinado.	
Acciones del Usuario	Respuesta del Sistema
1. El administrador accede al bloque de administración y selecciona la funcionalidad "Recolector".	1.1 El sistema muestra una página con varias opciones ("Listar usuarios pasivos" y "Listar actividades y recursos no utilizados").
2. El administrador elige la opción "Listar actividades y recursos no utilizados".	2.1 El sistema muestra una página con una tabla en la cual se encuentran todas las actividades o recursos por cada curso que hace un tiempo determinado por el administrador no son utilizados.
Resultado esperado: La lista de actividades y recursos no utilizados es mostrada de forma correcta en dependencia del tiempo de inactividad determinado por el administrador de la plataforma.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 31: Prueba de aceptación 5.

Caso de Prueba de Aceptación	
Código: HU6_P1	Historia de usuario: 6
Nombre Historia de usuario: Enviar notificación a los profesores editores de un curso	
Descripción: Prueba para la funcionalidad, enviar notificación a los profesores editores de un curso.	
Condiciones de ejecución: Debe existir al menos una actividad o recurso inactivo de un curso determinado.	
Acciones del Usuario	Respuesta del Sistema
1. El administrador accede al bloque de administración y selecciona la funcionalidad "Recolector".	1.1 El sistema muestra una página con varias opciones ("Listar usuarios pasivos" y "Listar actividades y recursos no utilizados").
2. El administrador elige la opción "Listar actividades y recursos no utilizados".	2.1 El sistema muestra una página con una tabla en la cual se encuentran todas las actividades y recursos por cada curso, que hace un tiempo determinado por el administrador no son utilizados.
3. El administrador selecciona cualquiera de las siguientes acciones: a) Seleccionar el botón "Seleccionar todo".	

b) Seleccionar solo las actividades y recursos a los cuales se desea enviar la notificación de correo al profesor editor.	
4. El administrador selecciona la opción “Enviar notificación”.	4.1 El sistema muestra una página donde se podrá redactar el correo que será enviado al profeso editor del curso.
5. El administrador da clic en el botón “Vista Previa”.	5.1 El sistema muestra una vista previa del correo que será enviado.
6. El administrador da clic en el botón “Enviar”.	6.1 El sistema envía la notificación de correo.
Curso Alternativo	
5. El administrador da clic en el botón “Cancelar”.	5.1 El sistema vuelve a la página principal.
Resultado esperado: La notificación de correo es enviada de forma correcta a los profesores editores del curso.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 32: Prueba de aceptación 6.

Caso de Prueba de Aceptación	
Código: HU7_P1	Historia de usuario: 7
Nombre Historia de usuario: Eliminar actividades y recursos inactivos	
Descripción: Prueba para la funcionalidad, eliminar actividades y recursos inactivos.	
Condiciones de ejecución: Debe existir al menos una actividad o recurso inactivo.	
Acciones del Usuario	Respuesta del Sistema
1. El administrador accede al bloque de administración y selecciona la funcionalidad “Recolector”.	1.1 El sistema muestra una página con varias opciones (“Listar usuarios pasivos” y “Listar actividades y recursos no utilizados”).
2. El administrador elige la opción “Listar actividades y recursos no utilizados”).	2.1 El sistema muestra una página con una tabla en la cual se encuentran todas las actividades y recursos por cada curso, que hace un tiempo determinado por el administrador no son utilizados.
3. El administrador selecciona cualquiera de las siguientes acciones: a) Seleccionar el botón “Seleccionar todo”. b) Seleccionar solo las actividades y recursos a los cuales se desea eliminar de la plataforma.	
4. El administrador selecciona la opción “Borrar”.	4.1 El sistema muestra una página donde se encuentran dos botones “Continuar” y

	"Cancelar".
5. El administrador da clic en el botón "Continuar".	5.1 El sistema elimina las actividades y recursos por cada curso inactivos de la plataforma.
Curso Alternativo	
5. El administrador da clic en el botón "Cancelar".	5.1 El sistema vuelve a la página principal.
Resultado esperado: Las actividades y recursos pasivos por cada curso son eliminados por completo de la plataforma de manera correcta.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 33: Prueba de aceptación 7.

3.2.2 Resultados obtenidos de las pruebas

Para la realización de estas pruebas se utilizaron las descripciones de los casos de prueba diseñados en el epígrafe 3.2.1. Con el propósito de detectar la mayor cantidad de errores posibles en la funcionalidad "Recolector" se realizaron tres iteraciones de pruebas. Estos errores se definen como No Conformidades (NC).

Las NC se clasificaron en Significativas, No Significativas y en Recomendaciones. Entiéndase por Significativa aquellas NC que puedan afectar el funcionamiento de la funcionalidad. No Significativas las enfocadas en el diseño u otro aspecto que no afecte el funcionamiento de la propuesta de solución. Recomendación, las sugerencias emitidas por los probadores.

El resultado obtenido se evidencia en el gráfico que se muestra a continuación, donde se puede observar la cantidad de No Conformidades detectadas en cada iteración.

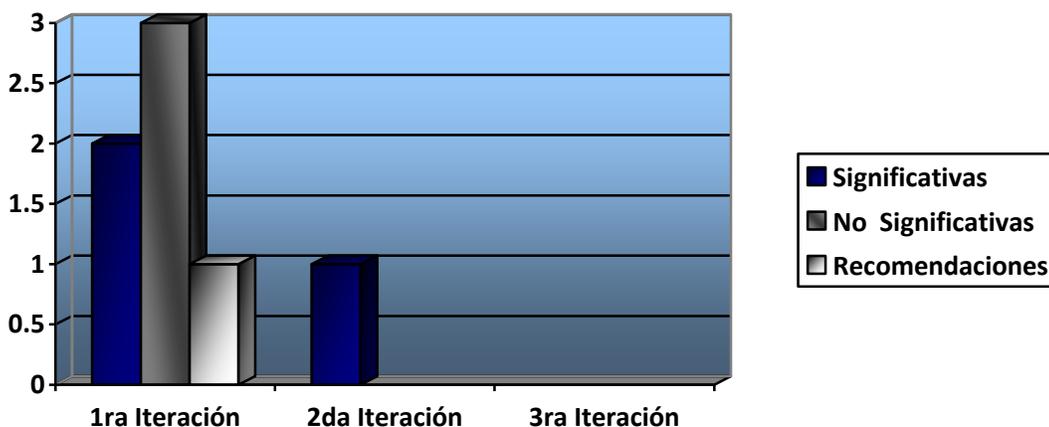


Ilustración 1: Gráfico para las No Conformidades identificadas en cada iteración de las pruebas.

Implementación y Prueba

Como se evidencia en la gráfica anterior se detectaron 6 NC en la primera iteración y 1 en la segunda, las cuales fueron resueltas satisfactoriamente. En la tercera iteración se evidenció el buen funcionamiento de la funcionalidad “Recolector” pues no se detectó ninguna NC. (Ver Anexo 4)

Conclusiones parciales

La funcionalidad “Recolector” se desarrolló teniendo en cuenta las tareas de ingeniería correspondientes para dar solución a las Historias de usuario.

Se obtuvo como resultado una funcionalidad que satisface los requerimientos planteados.

Los resultados obtenidos en las pruebas de aceptación luego de tres iteraciones garantizan la conformidad del cliente con la solución.

Conclusiones

Una vez finalizada la investigación y desarrollo del presente trabajo se concluye que:

- La no existencia de una funcionalidad para gestionar la información no utilizada en *Moodle* así como en los LMS analizados, confirmó la necesidad de la implementación de la funcionalidad “Recolector” para *Moodle* en su versión 2.3.x.
- La utilización de la metodología, los lenguajes y las herramientas más adecuadas para la implementación de la funcionalidad “Recolector”, según las características de *Moodle* y su arquitectura, permitió un desarrollo rápido y eficaz de la solución.
- Durante las fases de exploración y planificación se generaron los artefactos propuestos por la metodología XP tales como: Historias de usuario, Tarjetas CRC, Plan de entrega y Plan de iteraciones, los cuales documentan la solución y favorecen su calidad.
- La funcionalidad “Recolector” favorece la toma de decisiones por parte de los administradores del EVA a la hora de liberar espacio en la base de datos, además de proporcionar un mejor control sobre los usuarios inactivos y las actividades y recursos no utilizados en la plataforma.
- Las pruebas de aceptación realizadas permitieron comprobar el grado de satisfacción del cliente y demostrar que la funcionalidad “Recolector” está lista para su uso.

Recomendaciones

- Convertir la funcionalidad “Recolector” en un bloque para la plataforma de teleformación *Moodle* que se instale y desinstale de forma automática.
- Incorporar funcionalidades al “Recolector” de modo que sea capaz de gestionar toda la información no utilizada dentro de la plataforma.
- Tener en cuenta las siguientes validaciones:
 1. Implementar la funcionalidad “Determinar tiempo de inactividad límite” de forma tal que se distinga entre el tiempo de inactividad de los usuarios y las actividades o recursos no utilizados dentro de la plataforma.
 2. Verificar antes de eliminar un usuario inactivo que el mismo haya recibido una notificación de correo, con el objetivo de evitar errores humanos.
 3. Calcular y mostrar el tiempo de inactividad de los usuarios, actividades y recursos no utilizados en la plataforma.

Bibliografía

Accogli, Juan Ignacio. 2006. Ventajas del e-learning. [En línea] 14 de septiembre de 2006. [Citado el: 2 de diciembre de 2012.] http://www.mental-gym.com/Docs/ARTICULO_80.pdf.

Achour, Mehdi. 1997-2013. Manual de PHP. [En línea] 1997-2013. [Citado el: 20 de noviembre de 2012.] http://in2.php.net/distributions/manual/php_manual_es.html.gz.

Allende, Roberto. 2006. *Desarrollo de Portales y Extranet con Plone*. 2006.

Álvarez, Miguel Angel. 2011. DesarrolloWeb.com. [En línea] 10 de enero de 2011. [Citado el: 20 de diciembre de 2012.] <http://www.desarrolloweb.com/articulos/que-es-xhtml.html>

Angulo, Joseba. 2006. *La formación sin distancia*. 2006.

Beck, Kent. 2000. *Extreme Programming Explained*. 2000.

Beck, Kent. 2012. Metodologías SCRUM y XP. [En línea] 2012. [Citado el: 10 de diciembre de 2012.] <http://www.metodologíasSCRUMyXP-WikiUDO.htm>

Biscay, Carlos Emilio. 2010. *Los estandares de Palermo*. Universidad de Palermo : s.n., 2010.

Cabrera, Yandris Mata. 2011. *Repositorio de Objetos de Aprendizaje (RHODA 2.0)*. Habana : s.n., 2011.

Calle, Fani. 2011. Arquitectura 3 capas. [En línea] 2011. [Citado el: 18 de diciembre de 2012.] <http://www.slideshare.net/Decimo/arquitectura-3-capas>.

Casas, Sandra y Reinaga, Héctor. 2008. Identificación y Modelado de Aspectos Tempranos dirigido por Tarjetas de Responsabilidades y Colaboraciones. [En línea] 2008. [Citado el: 10 de diciembre de 2012.] <http://www.oocities.org/espanol/profeprog2/INVPAPER25.pdf>.

Clark, James. 2012. W3C. W3C Interaction domain. [En línea] enero de 2012. [Citado el: 16 de diciembre de 2012.] <http://www.w3.org>.

Favier, Adriana. 2010. Plataforma Virtual Dokeos. [En línea] 2010. [Citado el: 1 de diciembre de 2012.]

<http://repositoral.cuaed.unam.mx:8080/jspui/bitstream/123456789/1370/1/2005-04-02494Dokeos.pdf>.

Fowler. Martin, Guo, W., W.M. Healy, and M. Zhou. 2010. ZigBee-wireless mesh networks for building automation and control. in Networking, Sensing and Control (ICNSC), International Conference on. 2010.

Guo, W. and M. Zhou. 2009. An emerging technology for improved building automation control. in Sys-tems, Man and Cybernetics. SMC 2009. IEEE International Conference on. 2009.

Gutiérrez Gómez, Yordani y Hidalgo Guillén, Jesús. 2007. *Herramientas para la Reutilización de Contenidos a partir de la plataforma Moodle*. Habana : s.n., 2007.

Jacobson, Ivar, Booch, Grady y James, Rumbaugh. 2000. *El lenguaje unificado de modelado*. 2000.

Joskowicz, José. 2008. *Reglas y Prácticas en eXtreme Programming*. España : Universidad de Virgo, 2008.

Moodle. 2007. Moodle.org. [En línea] 21 de febrero de 2007. [Citado el: 4 de diciembre de 2012.] [http://docs.moodle.org/es/Usuarios en l%C3%ADnea](http://docs.moodle.org/es/Usuarios_en_l%C3%ADnea).

Moodle. 2011. Moodle.org. [En línea] 5 de enero de 2011. [Citado el: 4 de diciembre de 2012.] [http://docs.moodle.org/es/Notas de Moodle 2.0](http://docs.moodle.org/es/Notas_de_Moodle_2.0).

Negrín, León y Rojas, Fonseca. 2011. *INCORPORACIÓN DEL ESTÁNDAR SCORM AL BLOQUE C2SCORM PARA LA EXPORTACIÓN DE CURSOS A PAQUETES*. La habana : s.n., 2011.

Pech, Carlos. 2008. Moodle.org. [En línea] 20 de junio de 2008. [Citado el: 4 de diciembre de 2012.] <http://docs.moodle.org/es/Cursos.Moodle.org>.

Perez, José Ramón Gómez. 2010. Las TIC en educación. [En línea] 2 de marzo de 2010. [Citado el: 12 de noviembre de 2012.] <http://boj.pntic.mec.es/jgomez46/ticedu.htm>.

- Pressman, Roger S. 2002.** *Ingeniería de Software, un enfoque práctico.* 2002.
- Pressman, Roger S. 2009.** *Software Engineering.* s.l. : Séptima edición, 2009.
- Presso, Maria Isabel. 2011.** Aprendizaje en red comunidad moodle-presso. [En línea] 2011.<http://www.slideshare.net/mariapresso66/aprendizaje-en-red-comunidad-moodlepresso#btnNext>.
- Pupo, Reinier Alcayde. 2011.** *Diseño e implementación de un módulo para la integración de Moodle con un sistema de mensajería instantánea.* Universidad de las Ciencias Informáticas : s.n., 2011.
- Rodríguez, Msc. Dayra Iris Hechavarría. 2012.** manografias.com. [En línea] 2012. [Citado el: 18 de enero de 2013.] <http://www.monografias.com/trabajos92/gestion-requisitos/gestion-requisitos.shtml>.
- Romani, Jorge. 2012.** Plataformas virtuales de aprendizaje opensource I. [En línea] 2 de noviembre de 2012. [Citado el: 1 de diciembre de 2012.] <http://www.xarxatic.com/plataformas-virtuales-de-aprendizaje-opensource-i/>.
- Ruiz Ortiz , Dra. Lidia y Valdés Rodríguez, MSc. 2011.** Formación Postgraduada. [En línea] 20 de noviembre de 2011. [Citado el: 2 de diciembre de 2012.] <http://evapostgrado.uci.cu/mod/resource/view.php?id=8210>.
- Ruz, Fidel Castro. 1960.** Discurso pronunciado por el comandante Fidel Castro Ruz, en el acto celebrado por la sociedad Espeleologica de Cuba, en la Academia de Ciencias . [En línea] 15 de enero de 1960. [Citado el: 15 de mayo de 2013.] <http://www.cuba.cu/gobierno/discursos/1960/esp/f150160e.html>.
- Sáenz, Freddy E. Silva. 2011.** Desarrollo de Aplicaciones en Internet. [En línea] 2011. [Citado el: 22 de noviembre de 2012.] <http://lia.unet.edu.ve/Dai/Ddain/Intro.html>.
- Tarruella, Enrique Castro López. 2010.** *Moodle: Manual del profesor.* Dpto. de Bioquímica Biología Molecular y Fisiología, ULPGC : s.n., 2010.
- Torre, A. 2006.** Introducción a la plataforma Moodle. [En línea] 2006. [Citado el: 4 de diciembre de 2012.]

Glosario de términos

Funcionalidad: Conjunto de líneas de código que realizan una tarea específica y puede retornar un valor. Las funciones pueden tomar parámetros que modifiquen su funcionamiento. Las funciones son utilizadas para descomponer grandes problemas en tareas simples y para implementar operaciones que son comúnmente utilizadas durante un programa y de esta manera reducir la cantidad de código.

Módulo: Porción de un programa de computadora. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas (o varias, en algún caso).

Multiplataforma: Es un término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

Software Libre: Significa que el software respeta la libertad de los usuarios y la comunidad. En términos generales, los usuarios tienen la libertad de copiar, distribuir, estudiar, modificar y mejorar el software. Con estas libertades, los usuarios (tanto individualmente como en forma colectiva) controlan el programa.

PDF (Formato de Documento Portátil): Formato de almacenamiento de documentos multiplataforma (Microsoft Windows, Unix, Mac) desarrollado por la empresa Adobe System. Especialmente ideado para documentos susceptibles de ser impresos.

Plataforma educativa: Son aplicaciones que facilitan la creación de entornos de enseñanza-aprendizaje, integrando materiales didácticos y herramientas de comunicación, colaboración y gestión educativa.

XML (lenguaje de marcas extensible): Metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Permite definir la gramática de lenguajes específicos, por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

HTML (lenguaje de marcas hipertextuales): Lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Mahara: Es una aplicación web en código abierto para gestionar ePortfolio y Redes sociales. Ofrece a los usuarios herramientas para crear y mantener un portafolio digital sobre su formación.

Flickr: Es un sitio web gratuito que permite almacenar, ordenar, buscar, vender y compartir fotografías y videos en línea.

Sun Microsystems: Fue una empresa informática que se dedicaba a vender estaciones de trabajo, servidores, componentes informáticos, software (sistemas operativos) y servicios informáticos.

Anexos

Anexo 1: Punto de partida para la selección de la metodología, lenguajes y herramientas

Investigaciones donde se realiza un estudio comparativo entre diferentes metodologías de desarrollo, lenguajes y herramientas y se seleccionan las más adecuadas para el desarrollo de plugins, funcionalidades y módulos para el LMS *Moodle*:

1. Título: Arquitectura de la Herramienta Andrómeda.

Autores: Pablo Molina Toledo y Henry Rodríguez Estévez.

Año: 2012.

Descripción: Se define una arquitectura de software haciendo usos de estilos y patrones arquitectónicos que permitan el desarrollo y mantenimiento de la herramienta Andrómeda, facilitando el diseño del despliegue de WSAE en entornos interiores.

Criterio de Referencia: Para seleccionar la metodología de software a utilizar en la implementación de la funcionalidad "Recolector".

Referencia: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_05537_12

2. Título: Portal Web para la Gestión de la Información del Artista Plástico.

Autores: Mara Morales Alfonso.

Año: 2012.

Descripción: Se elabora un Portal Web para la gestión de la información de la obra del artista plástico mediante el uso de la tecnología Web 2.0. Se realiza un estudio para seleccionar las herramientas más factibles, teniendo en cuenta las tendencias del desarrollo web. Se analizan algunos de los sistemas similares definiendo posibles funcionalidades para la solución propuesta, agregándole las necesidades del cliente.

Criterio de Referencia: Para seleccionar la metodología de software a utilizar en la implementación de la funcionalidad "Recolector".

Referencia: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_05566

3. Título: Optimización de la generación de estadísticas en la plataforma de teleformación *Moodle* 1.9.x.

Autores: Celia Miranda Fernández y Víctor Manuel Álvarez Castillo.

Año: 2012.

Descripción: Se elabora un módulo para el Entorno Virtual de Aprendizaje, el cual facilita la generación automática de datos estadísticos. El objetivo de esta funcionalidad radica en optimizar la generación de estadísticas en la versión 1.9.x de la plataforma de teleformación Moodle.

Criterio de Referencia: Para seleccionar los lenguajes y herramientas a utilizar en la implementación de la funcionalidad "Recolector".

Referencia: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_05551_12

4. Título: Bloque de exportación de contenidos en *Moodle* 1.9.x para dispositivos móviles.

Autores: Adrián González Peraza y Danichel Morejón Roque.

Año: 2012.

Descripción: Se elabora el bloque CTEBOOK (Course To E-Book), el cual permite exportar el contenido de los cursos de la plataforma a libro electrónico interactivo. Con este bloque los usuarios pueden obtener un recurso portable que facilita la búsqueda y enriquece el texto a través de enlaces multimedia, haciendo que el contenido de los cursos que se encuentran en un entorno virtual sea leído con facilidad.

Criterio de Referencia: Para seleccionar los lenguajes y herramientas a utilizar en la implementación de la funcionalidad "Recolector".

Referencia: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_05575_12

Anexo 2: Entrevista realizada a los administradores de la plataforma de teleformación *Moodle* en la UCI

Objetivo: Conocer las principales dificultades de los administradores al interactuar con la información no utilizada dentro de la plataforma.

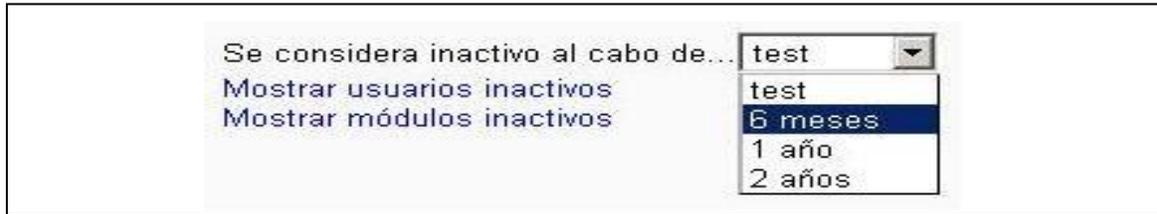
Entrevistado: Técnico en Ciencias Informáticas Levian Diaz Camejo.

La entrevista estuvo guiada por las siguientes preguntas:

- ¿Cómo se puede determinar si un usuario o una actividad o recurso de un curso no están siendo utilizados en la plataforma?
- ¿Cómo se calcula o define el tiempo que está inactivo un usuario o una actividad o recurso de un curso en la plataforma?
- ¿Cómo se gestionan los usuarios y cursos actualmente en la plataforma?

Anexo 3: Interfaces

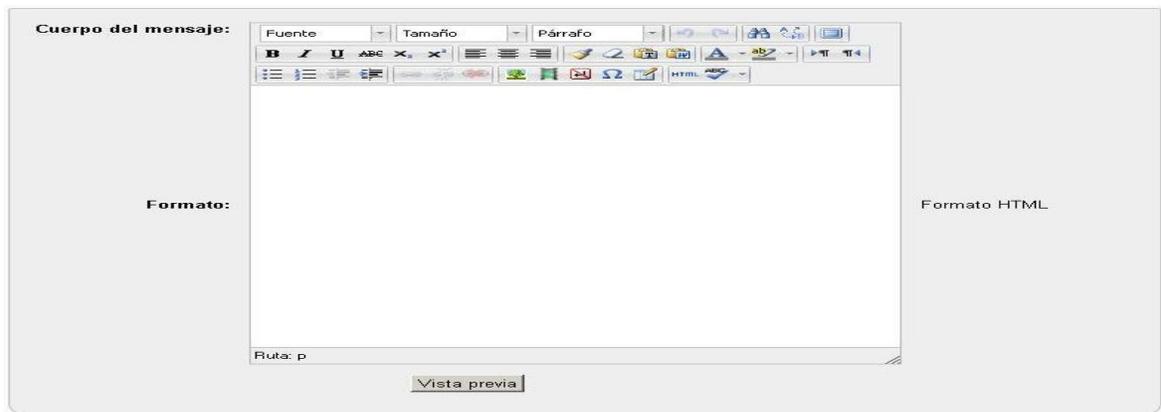
Interfaz para determinar el tiempo de inactividad de los usuarios inactivos y actividades o recursos no utilizados



Interfaz para listar usuarios inactivos



Interfaz para enviar mensaje a los usuarios inactivos



Interfaz para eliminar usuarios inactivos

Eliminar usuarios

¿Desea eliminar los usuarios seleccionados?

Interfaz para listar módulos no utilizados

Buscar cursos:

curso1 [Notificar profesor](#)

Seleccionar	Actividad	Vistas	Entradas de blog relacionadas	Último acceso
<input type="checkbox"/>	 Novedades	-	-	Nunca
<input type="checkbox"/>	 Foro	-	-	Nunca

Con los módulos seleccionados...

Interfaz para enviar notificación a los profesores editores de un curso donde los módulos no son utilizados

Cuerpo del mensaje:

Fuente Tamaño Párrafo

B *I* U ABC x₂ x₃          

Los siguientes módulos se encuentran inactivos en el curso c1:

 **Tarea** Último acceso: martes, 28 de mayo de 2013, 05:19 (3 minutos 25 segundos)

 **Encuesta** Último acceso: martes, 28 de mayo de 2013, 05:19 (3 minutos 24 segundos)

Ruta: p

Formato HTML

Usuarios seleccionados		
Marlon Campa Matos	mcampa@estudiandes.uci.cu	<input type="button" value="Quitar"/>
Augusto Marquez Ramos	amarquez@estudiandes.uci.cu	<input type="button" value="Quitar"/>
Dianne Feriol Alvarez	dferiol@estudiandes.uci.cu	<input type="button" value="Quitar"/>

Interfaz para la vista previa del mensaje que será enviado a los profesores editores de un curso donde los módulos no son utilizados.

Vista previa formato HTML

Los siguientes módulos se encuentran inactivos en el curso c1:

-  Tarea Último acceso: martes, 28 de mayo de 2013, 05:19 (3 minutos 25 segundos)
- Encuesta Último acceso: martes, 28 de mayo de 2013, 05:19 (3 minutos 24 segundos)

Interfaz para eliminar los módulos no utilizados

Eliminar módulos

¿Desea eliminar los módulos seleccionados?

Anexo 4: Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Clasificación
Recolector	1	Error de traducción	Palabra "Profesor" sin traducir.	1ra iteración	NS
Recolector	2	Error de traducción	Palabra "Recolector" sin traducir.	1ra iteración	NS
Recolector	3	Error de traducción	Signo "?" sin traducir.	1ra iteración	NS
Recolector	4	Botón no funcional	Botón "Quitar" no funcional.	1ra iteración	S
Recolector	5	Botón no funcional	Botón "Borrar" no funcional.	1ra y 2da iteración	S
Recolector	6	Interfaz Gráfica	Problemas en la interfaz de la funcionalidad listar usuarios inactivos.	1ra iteración	R

Tabla 34: Registro de defectos y dificultades detectados.