

**ESCUELA POLITÉCNICA DEL EJÉRCITO
EXTENSIÓN LATACUNGA**



CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**“DESARROLLO DE UN PROTOTIPO DE AULA VIRTUAL
(LEARNING MANAGEMENT SYSTEM) UTILIZANDO OPEN
SOURCE PARA LA MAESTRIA EN INGENIERIA DE
SOFTWARE DEL AREA DE POSGRADOS DE LA ESCUELA
POLITÉCNICA DEL EJÉRCITO EXTENSIÓN LATACUNGA”**

MARCO VINICIO PROAÑO VASCO

TESIS PRESENTADA COMO REQUISITO PREVIO A LA
OBTENCIÓN DEL GRADO DE INGENIERA EN SISTEMAS E
INFORMÁTICA

AÑO 2011

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por Marco Vinicio Proaño Vasco, bajo nuestra supervisión.

Ing. Fabián Montaluisa
DIRECTOR DE PROYECTO

Ing. Javier Montaluisa
CODIRECTOR DE PROYECTO

Ing. José Luis Carrillo
DIRECTOR DE CARRERA

Dr. Rodrigo Vaca
SECRETARIO ACADÉMICO

DECLARACIÓN DE RESPONSABILIDAD

Yo: Marco Vinicio Proaño Vasco

DECLARO QUE:

El proyecto de grado denominado “DESARROLLO DE UN PROTOTIPO DE AULA VIRTUAL (LEARNING MANAGEMENT SYSTEM) UTILIZANDO OPEN SOURCE PARA LA MAESTRIA EN INGENIERIA DE SOFTWARE DEL AREA DE POSGRADOS DE LA ESCUELA POLITÉCNICA DEL EJÉRCITO EXTENSIÓN LATACUNGA” ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, 21 de noviembre del 2011.

Marco Vinicio Proaño Vasco
CI: 1804013082

AUTORIZACIÓN

YO, MARCO VINICIO PROAÑO VASCO

Autorizo a la Escuela Politécnica del Ejército, la publicación en la biblioteca virtual de la Institución, el trabajo: “DESARROLLO DE UN PROTOTIPO DE AULA VIRTUAL (LEARNING MANAGEMENT SYSTEM) UTILIZANDO OPEN SOURCE PARA LA MAESTRIA EN INGENIERIA DE SOFTWARE DEL AREA DE POSGRADOS DE LA ESCUELA POLITÉCNICA DEL EJÉRCITO EXTENSIÓN LATACUNGA”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Latacunga, noviembre de 2011.

Marco Vinicio Proaño Vasco
CI: 1804013082

DEDICATORIA

A mis padres, Marco y Bety por su apoyo incondicional a lo largo de toda mi vida, y por enseñarme que con perseverancia se conquistan los más grandes y difíciles retos.

A mi hermana, Anita por creer siempre en mí y por tenerme como un ejemplo a seguir, esto es para ti.

A mi novia, Diana que con su cariño y comprensión a estado siempre en mis momentos de debilidad y apremio.

A mis amigos, gracias chicos por su ayuda y palabras de aliento en todo momento de este trayecto.

AGRADECIMIENTO

El agradecimiento más grande para mis padres por siempre estar ahí para mí en cualquier instante que los necesitara, incluso cuando en tramos de mi vida quise renunciar, gracias, los amo.

A la Escuela Politécnica del Ejército Extensión Latacunga, y a mis maestros en general por trazar la línea del conocimiento y rectitud para alcanzar todos los objetivos que me propusiera, tanto en lo personal como en lo académico.

A mis compañeros y amigos con quienes compartí las experiencias más enriquecedoras de mi existencia.

Al departamento de las TICS de la ESPE-L por toda su colaboración en la elaboración de este proyecto, en especial a la Ing. Mónica Llumitasig por su esfuerzo y dedicación en el desarrollo de la plataforma.

RESUMEN

La Escuela Politécnica del Ejército Extensión Latacunga al ser una de las instituciones con más prestigio del Ecuador, se ve en la obligación de estar siempre a la vanguardia en cuanto a sistemas de información y de apoyo a las necesidades que se le presentan diariamente al estudiante politécnico.

Este proyecto realizará el desarrollo de un prototipo de aula virtual (LMS) para la escuela y tiene como objetivo apoyar el proceso de enseñanza-aprendizaje en programas de pregrado y postgrado de la institución.

En el Capítulo I se seleccionará al LMS Open Source a través del establecimiento de criterios de selección y su comparación entre ellos. Además, se efectuará un análisis de la arquitectura, metodología, componentes o módulos, etc. del LMS seleccionado.

En el Capítulo II se tratará sobre la selección de la metodología de desarrollo, para ello se efectuará una descripción, análisis y comparación de dos metodologías más utilizadas, en base a la descripción del proyecto que se pretende resolver. Este mismo proceso se seguirá para las herramientas de desarrollo a utilizarse en el proyecto.

En el Capítulo III se identifica y detalla el problema en base a su definición y se introduce a la fase del análisis, en el cual se definen las historias de usuario, número de iteraciones, la planificación de entregas en base a estas, etc. Para la implementación se utiliza el desarrollo iterativo, por tal razón cada iteración contempla el proceso de las metodologías aplicadas, obteniendo así el producto final.

En el Capítulo IV se presentan las conclusiones y recomendaciones que han sido elaboradas una vez terminado el desarrollo de la plataforma.

ABSTRACT

“Escuela Politécnica del Ejército Extensión Latacunga” is one of the most prestigious institutions in Ecuador, so there begins the necessity to be always at the forefront of information systems and to support the needs that are presented daily to the polytechnic students.

This project will conduct the development of a virtual classroom (LMS) prototype for the school and aims to support the teaching and learning in undergraduate and graduate programs of the institution.

In chapter I, the Open Source LMS will be selected through the establishment of some selection criteria and comparison between them. In addition, there will be an analysis of the architecture, methodology, components or modules, etcetera of the selected LMS.

In Chapter II, there will be discussed the selection of a development methodology, there will be a description, analysis and comparison of the two most used methodologies, based on the project description to be resolved. This same process is followed for the development tools used in the project.

Chapter III identifies and details the problem based on its definition and is introduced to a stage of analysis, which defines the user stories, number of iterations, the release planning based on these, etcetera. Iterative development is used for the implementation, for this reason each iteration includes the process of the methodologies applied, leading this to obtain the final product.

Chapter IV presents the conclusions and recommendations that have been made once the platform development is finished.

ÍNDICE GENERAL

ABSTRACT.....	viii
ÍNDICE GENERAL	ix
ÍNDICE DE TABLAS.....	xii
ÍNDICE DE GRÁFICOS	xiv
ÍNDICE DE GRÁFICOS.....	xiv
CAPITULO 1.....	1
1. SITUACION ACTUAL DE LA MAESTRIA EN INGENIERIA EN SOFTWARE	1
1.1.- INTRODUCCIÓN.....	1
1.2.- ANTECEDENTES.....	2
1.3.- OBJETIVOS DE LA INVESTIGACIÓN	3
1.3.1.- OBJETIVO GENERAL.....	3
1.3.2.- OBJETIVOS ESPECÍFICOS	3
CAPÍTULO 2.....	5
2. SELECCIÓN DEL SISTEMA OPEN SOURCE – LMS (LEARNING MANAGEMENT SYSTEM)	5
2.1.- ESTABLECIMIENTO DE CRITERIOS DE SELECCIÓN	5
2.1.1.- CRITERIOS DE SELECCIÓN	6
2.1.1.1.- Funcionalidad.....	8
2.1.1.2.- Fiabilidad	8
2.1.1.3.- Facilidad de Uso	9
2.1.1.4.- Eficiencia	9
2.1.1.5.- Mantenimiento	9
2.1.1.6.- Portabilidad.....	9
2.2.- SELECCIÓN DEL LMS.....	10
2.2.1.- COMPARACIÓN DE LOS LMS	10
2.2.1.1.- Producto: Dokeos	11
2.2.1.2.- Producto: Moodle.....	18
2.2.1.3.- Producto: Atutor.....	26
2.3.- ANÁLISIS DE LA ARQUITECTURA Y METODOLOGÍA DEL LMS SELECCIONADO.....	36
2.3.1.- DESCRIPCIÓN DE MOODLE 1.9.x	36
2.3.2.- CRITERIOS PARA SU ARQUITECTURA.....	36
2.4.- CARACTERÍSTICAS DE MOODLE.....	39
2.4.1.- TÉCNICAS	39
2.4.2.- DISEÑO GENERAL.....	39
2.4.3.- ADMINISTRACIÓN DEL SITIO	40
2.4.4.- ADMINISTRACIÓN DE USUARIOS.....	41
2.4.5.- ADMINISTRACIÓN DE CURSOS	42
2.5.- IDENTIFICACIÓN Y DESCRIPCIÓN DE LOS MÓDULOS EXISTENTES.....	43
2.5.1.- MÓDULO DE TAREAS	44
2.5.2.- MÓDULO DE CHAT	44
2.5.3.- MÓDULO DE CONSULTA	44

2.5.4.-	MÓDULO DE FORO	45
2.5.5.-	MÓDULO CUESTIONARIO	45
2.5.6.-	MÓDULO DE RECURSO	46
2.5.7.-	MÓDULO DE ENCUESTA	47
2.5.8.-	MÓDULO DE TALLER	47
2.6.-	REQUERIMIENTOS FÍSICOS PARA PRESTAR SERVICIO EN LA ESPE-L	48
2.6.1.-	SERVIDOR.....	48
2.6.2.-	ESTACIONES DE TRABAJO.....	48
CAPÍTULO 3.....		50
3.	SELECCIÓN DE LA METODOLOGÍA Y LAS HERRAMIENTAS DE DESARROLLO	50
3.1.-	SELECCIÓN DE LA METODOLOGÍA.....	50
3.1.1.-	METODOLOGÍAS ÁGILES VS METODOLOGÍAS TRADICIONALES.....	50
3.1.1.1.-	Metodologías Tradicionales	50
3.1.1.2.-	Metodologías Ágiles.....	51
3.1.2.-	DESCRIPCIÓN DEL PROBLEMA.....	53
3.1.3.-	DESCRIPCIÓN DE LA METODOLOGÍA DE DESARROLLO: EXTREME PROGRAMMING (XP)	56
3.1.3.1.-	Prácticas de eXtreme Programming	57
3.1.3.2.-	Roles de Extreme Programming.....	61
3.1.3.3.-	Proceso de eXtreme Programming.....	63
3.1.4.-	DESCRIPCIÓN DE LA METODOLOGÍA DE DESARROLLO: SCRUM 66	
3.1.4.1.-	Características De Un Proceso Scrum	66
3.1.4.2.-	Proceso, Roles Y Reuniones De Scrum	67
3.1.5.-	COMPARATIVA DE LAS METODOLOGÍAS SCRUM Y XP	72
3.1.5.1.-	Semejanzas	72
3.1.5.2.-	Diferencias	73
3.1.6.-	SCRUM Y XP	74
3.1.7.-	PROCESO SCRUM CON PRÁCTICAS DE INGENIERÍA XP 75	
3.1.8.-	APLICACIÓN DE SCRUM Y EXTREME PROGRAMMING EN EL PROYECTO	76
3.1.8.1.-	Sprints y Sprint Planning Meetings.	77
3.1.8.2.-	Product Backlog y Product Owner	78
3.1.8.3.-	Sprint Backlog.....	78
3.1.8.4.-	Scrum Master.....	78
3.1.8.5.-	Diseño Simple.....	78
3.1.8.6.-	Pruebas - Tests.....	78
3.1.8.7.-	Integración Continua.....	79
3.1.8.8.-	Refactorización	79
3.1.8.9.-	Programación por Parejas	79
3.1.8.10.-	Releases Pequeños.....	79
CAPÍTULO 4.....		80
4.	DESARROLLO DEL SISTEMA.....	80
4.1.-	DEFINICIÓN DEL PROBLEMA	80

4.2.- ANALISIS (PREPARACIÓN DEL PROYECTO).....	81
4.2.1.- ROLES DEL PROCESO DE DESARROLLO	82
4.2.2.- HISTORIAS DE USUARIO (XP).....	82
4.2.2.1.- Historia 1: Creación de Carreras o Programas Académicos	83
4.2.2.2.- Historia 2: Visualización, Modificación, Eliminación y	
Búsqueda de Carreras o Programas Académicos.....	84
4.2.2.3.- Historia 3: Gestión de Cursos Académicos.....	85
4.2.2.4.- Historia 4: Gestión de Recursos del Curso	86
4.2.2.5.- Historia 5: Gestión de Registro de Estudiante	87
4.2.2.6.- Historia 6: Gestión de Calificaciones	88
4.2.2.7.- Historia 7: Gestión de Perfil de Usuario	89
4.2.2.8.- Historia 8: Diseño estético de la Plataforma	90
4.2.3.- PRODUCT BACKLOG (SCRUM)	91
4.3.- PLANIFICACIÓN DE ENTREGAS (SPRINT PLANNING).....	92
4.3.1.- HERRAMIENTAS	94
4.3.1.1.- Desarrollo	94
4.4.- DESARROLLO ITERATIVO	94
4.4.1.- ITERACIÓN (SPRINT) 1	95
4.4.1.1.- Sprint Backlog.....	95
4.4.1.2.- Revisión del Sprint.....	96
4.4.2.- ITERACIÓN (SPRINT) 2	102
4.4.2.1.- Sprint Backlog.....	102
4.4.2.2.- Revisión del Sprint.....	103
4.4.3.- ITERACIÓN (SPRINT) 3	107
4.4.3.1.- Sprint Backlog.....	107
4.4.3.2.- Revisión del Sprint.....	108
4.4.4.- ITERACIÓN (SPRINT) 4	112
4.4.4.1.- Sprint Backlog.....	112
4.4.4.2.- Revisión del Sprint.....	113
4.4.4.3.- Product Backlog Final	117
4.5.- PRUEBAS	118
4.5.1.- PLAN DE PRUEBAS	118
CAPITULO 5.....	144
5. CONCLUSIONES Y RECOMENDACIONES.....	144
5.1.- CONCLUSIONES.....	144
5.2.- RECOMENDACIONES.....	146
GLOSARIO DE TÉRMINOS	147
REFERENCIAS BIBLIOGRAFICAS	150
Libros	150
Proyectos de Titulación	151
Web.....	151

ÍNDICE DE TABLAS

Tabla I.1. Ficha de Producto Dokeos.....	12
Tabla I.2. Ficha Técnica Dokeos	14
Tabla I.3. Ficha del Producto Moodle	19
Tabla I.4. Ficha Técnica Moodle	21
Tabla I.5. Ficha del producto Atutor	27
Tabla I.6. Ficha Técnica Atutor.....	29
Tabla II.1. Comparativa Metodologías Ágiles vs. Tradicionales.....	52
Tabla II.2. Diferencias SCRUM y Xp.....	73
Tabla III.1. Historia 1 - Creación de carreras o programas académicos ...	83
Tabla III.2. Historia 2 - Visualización, modificación, eliminación y búsqueda de carreras o programas académicos	84
Tabla III.3. Historia 3 - Gestión de Cursos Académicos	85
Tabla III.4. Historia 4 - Gestión de Recursos del Curso.....	86
Tabla III.5. Historia 5 - Gestión de Registro de Estudiante.....	87
Tabla III.6. Historia 6 - Gestión de Calificaciones.....	88
Tabla III.7. Historia 7 – Gestión de perfil de usuario	89
Tabla III.8. Historia 8 – Gestión de Récord Académico por Curso	90
Tabla III.9. Sinopsis de las Historias de usuarios	93
Tabla III.10. Sprint Planning.....	93
Tabla III.11. Sprint Backlog – Iteración 1.....	95
Tabla III.12. Sprint Backlog – Iteración 2.....	102
Tabla III.13. Sprint Backlog - Iteración 3.....	107
Tabla III.14. Sprint Backlog - Iteración 4.....	112
Tabla III.15.Tabla 3.15 Prueba de registro de un nuevo usuario	118
Tabla III.16. Prueba de Eliminación de Usuarios.....	119
Tabla III.17. Prueba de Inicio de Sesión.....	120
Tabla III.18. Prueba de Creación de Carrera o Programa Académico....	121
Tabla III.19. Prueba de Listado de Carreras	122
Tabla III.20. Prueba de Visualización de Carreras	123
Tabla III.21. Prueba de Modificación de Carreras	124
Tabla III.22. Prueba de Eliminación de Carreras.....	125
Tabla III.23. Prueba de creación de cursos en una carrera o programa académico.....	126
Tabla III.24. Prueba de dada de baja a los cursos que pertenecen a una carrera o programa académico.....	127
Tabla III.25. Prueba de modificación de un curso perteneciente a una carrera o programa académico.....	128
Tabla III.26. Prueba de Asignación de Roles Globales.....	129
Tabla III.27. Prueba de Asignación de Roles (Curso)	131
Tabla III.28. Prueba de Creación de Sala de Chat.....	132
Tabla III.29. Prueba de Utilización de Sala de Chat.....	133
Tabla III.30. Prueba de Creación de Foros	134
Tabla III.31. Prueba de Ingreso e Interacción en Foros.....	135
Tabla III.32. Prueba de Carga de Material de Consulta (Profesor).....	136

Tabla III.33. Prueba de Creación de una Tarea (Profesor)	138
Tabla III.34. Prueba de Interacción (Consulta, Resolución) en Tareas...	139
Tabla III.35. Prueba de Creación de Cuestionarios Calificados (Profesor)	140
Tabla III.36. Prueba de Rendición de Cuestionarios Calificados (Estudiante)	141
Tabla III.37. Prueba de Consulta de Calificaciones (Estudiante).....	142

ÍNDICE DE GRÁFICOS

Gráfico I.1. Comparación cuantitativa de LMS.....	35
Gráfico I.2. Comparación de LMS	35
Gráfico II.1. Diagrama de eXtreme Programming	56
Gráfico II.2. Prácticas de eXtreme Programming	61
Gráfico II.3. Proceso de SCRUM.....	70
Gráfico II.4. Algunas Reuniones de SCRUM	72
Gráfico III.1. Gráfica de Grantt Plataforma Virtual ESPE-L	91
Gráfico III.2. Tareas Completadas Marco - Iteración 1	96
Gráfico III.3. Tareas Completadas Mónica - Iteración 1	97
Gráfico III.4. Tareas Pendientes Marco - Iteración 1	98
Gráfico III.5. Tareas Pendientes Mónica - Iteración 1	99
Gráfico III.6. Burn Down Chart - Iteración 1	100
Gráfico III.7. Estado de Tareas - Iteración 1	101
Gráfico III.8. Tareas Completadas Marco - Iteración 2	103
Gráfico III.9. Tareas Completadas Mónica - Iteración 2.....	104
Gráfico III.10. Tareas Pendientes Marco - Iteración 2.....	105
Gráfico III.11. Tareas Pendientes Mónica - Iteración 2	105
Gráfico III.12. Burn Down Chart - Iteración 2	106
Gráfico III.13. Estado de Tareas - Iteración 2	106
Gráfico III.14. Tareas Completadas Marco - Iteración 3	108
Gráfico III.15. Tareas Completadas Mónica - Iteración 3	109
Gráfico III.16. Tareas Pendientes Marco - Iteración 3.....	110
Gráfico III.17. Tareas Pendientes Mónica - Iteración 3	110
Gráfico III.18. Burn Down Chart - Iteración 3	111
Gráfico III.19. Estado de Tareas - Iteración 3	111
Gráfico III.20. Tareas Completadas Marco - Iteración 4	113
Gráfico III.21. Tareas Completadas Mónica - Iteración 4	114
Gráfico III.22. Burn Down Chart - Iteración 4	115
Gráfico III.23. Estado de Tareas - Iteración 4	116
Gráfico III.24. Gráfico 3.24. Product Backlog Final.....	117

CAPITULO 1.

1. SITUACION ACTUAL DE LA MAESTRIA EN INGENIERIA EN SOFTWARE

1.1.- INTRODUCCIÓN

La Escuela Politécnica del Ejército Extensión Latacunga al ser una de las instituciones con más prestigio del Ecuador, se ve en la obligación de estar siempre a la vanguardia en cuanto a sistemas de información y de apoyo a las necesidades que se le presentan diariamente al estudiante politécnico.

Los estudiantes de posgrado de la ESPE extensión Latacunga requieren de un sistema específico para el desarrollo de las actividades académicas programadas en cada una de las especialidades que ofrece el área, que les permita solventar de manera rápida y oportuna sus necesidades de investigación y producción académica con la implementación de un prototipo de sistema Open Source de Aula virtual que agilite el proceso académico e investigativo.

De esta manera el sistema de ejecución de cada uno de los programas de aprendizaje semi-presenciales con la ayuda de las TI alcanzaría un nivel académico más eficiente, logrando de esta manera calidad y excelencia en la educación superior.

Este sistema Open Source se convertirá en un método de intercambio de información por medio de las tecnologías de la información entre el profesor y los alumnos.

Es importante resaltar que hay que buscar que el producto final (implementación del aula virtual) utilizando el sistema Open Source, sea

fácil de manejar por parte de estudiantes y docentes no especializados en sistemas (no se necesita programación) y que se puedan utilizarlo en un amplio rango de aplicaciones educativas (su uso no se restringe a un área del saber específica, se puede utilizar para dictar cualquier curso).

1.2.- ANTECEDENTES

En la actualidad, las universidades que cuentan con ofertas académicas necesarias para satisfacer las necesidades de una comunidad tratan de incorporar, entre los servicios que brindan a sus alumnos, la opción de que puedan recibir su educación vía Internet (video conferencias, clases virtuales, e-mail, foros de discusión, seminarios, etc.). Todas estas aplicaciones deberán ser colocadas dentro de una plataforma de software exclusiva de e-learning, que es un tipo de enseñanza en tiempo real, con un instructor o autoadministrado, mismas que hacen posible la actualización constante de habilidades y el aprendizaje permanente de nuevas prácticas, gracias al uso de computadoras interconectadas entre sí.

Una de las características básicas del e-learning es su enfoque centrado en el usuario. A diferencia de la formación tradicional en la que, bien el tutor o bien el contenido son los elementos centrales, el e-learning pone al usuario en el centro del proceso de aprendizaje, convirtiéndolo en motor y protagonista de su propia experiencia educativa

Con los crecientes avances tecnológicos estamos inmersos en una sociedad de constantes cambios; en la que la vida virtual encaja cada vez más en nuestra vida personal, a tal punto que muchas actividades cotidianas comienzan a realizarse a través de Internet, induciendo así al uso de las nuevas tecnologías, creando el mayor suceso de nuestros tiempos conocido como Internet, así, se podría decir que el desarrollo de la tecnología ha cambiado tanto el estilo de vida de las personas, que una

computadora en cada hogar es ahora tan necesaria como una cocina o un refrigerador. La mayor parte de actividades se desarrollan desde un computador; y es por eso que se ha difundido de manera predominante el concepto de intercambio de información a través de la Web como canal de difusión, por lo que se ha considera necesario la implementación de esta Aula Virtual para facilitar a los estudiantes una herramienta de fácil acceso a la información.

Teniendo en cuenta lo expuesto, es posible decir que la creación son fines académicos de esta aula virtual para la Maestría en Ingeniería en Software, es una opción acertada en estos días; y, utilizando el proceso de difusión de índole viral y el alcance masivo a favor, es factible formar comunidades académicas en poco tiempo, en las que las personas podrían publicar sus dudas, comentarios, sugerencias y conocimiento en general.

1.3.- OBJETIVOS DE LA INVESTIGACIÓN

1.3.1.- OBJETIVO GENERAL

Desarrollo de un prototipo de aula virtual (LMS) utilizando Open Source para la Maestría en Ingeniería de Software del área de posgrados de la Escuela Politécnica Del Ejército Sede Latacunga.

1.3.2.- OBJETIVOS ESPECÍFICOS

- Estudiar el arte de las Aulas Virtuales
- Analizar los requerimientos de los estudiantes, docentes y autoridades del área de posgrados
- Elaborar el sistema Open Source de Aula Virtual con los cursos impartidos en el área de Posgrados de la escuela.

- Renovar la comunicación entre Profesor - Alumno, Alumno - Alumno, Profesor - Profesor

CAPÍTULO 2.

2. SELECCIÓN DEL SISTEMA OPEN SOURCE – LMS (LEARNING MANAGEMENT SYSTEM)

2.1.- ESTABLECIMIENTO DE CRITERIOS DE SELECCIÓN

El software libre aplicado a contextos educativos, ofrece posibilidades que pueden favorecer el proceso de enseñanza – aprendizaje en función de los destinatarios, de sus necesidades, su nivel de formación ya que puede ser modificado y adaptado de acuerdo a nuestros intereses y a los objetivos que persigamos.

Podemos definir un LMS como un sistema que organiza las actividades de formación dentro de una institución (JOIN¹, 2005). Las plataformas gestoras de aprendizaje o LMS, incluyen una variedad de herramientas y funcionalidades que es posible aplicar a cualquier de las aproximaciones de Blended Learning².

Los servicios de educación virtual no sólo consisten en gestionar un LMS, si no que constituyen una congruencia de elementos que abstraen en su conjunto la estructura organizacional y misional de la institución educativa (ESPE-L), por lo tanto la selección de un LMS debe cumplir con las necesidades académicas de la institución y como herramienta de apoyo al modelo pedagógico de educación virtual, de tal manera, se deben

1 JOIN - Proyecto europeo financiado por la iniciativa e-Learning de la Comisión Europea, que evalúa la calidad de LMS 1 open source para poder ofrecer información y apoyo a toda la comunidad que desee adoptar alguno de estos sistemas

2 Blended Learning: proceso docente semipresencial

considerar algunos elementos que apoyen la postura antes mencionada. Para ello es necesario que el LMS no sea una plataforma o recetario de contenidos, debe permitir modularlos; así como sus objetos de aprendizaje, debe también permitir gestionar recursos administrativos como herramientas de control de usuarios, seguimiento, valoración - evaluación, roles, reportes e informes teniendo en cuenta los estándares (SCROM)³ y normas que regulan esta forma de enseñanza mediada por las TIC, dando la oportunidad de migrarlas a otras plataformas, además de aspectos técnicos para la evaluación de un software según la norma ISO/IEC 9126⁴ como lo es la funcionalidad, la confiabilidad, la usabilidad, la eficiencia, mantenibilidad y portabilidad.

2.1.1.- CRITERIOS DE SELECCIÓN

El sistema de gestión de aprendizaje debe permitir la gestión y configuración de los elementos que componen un sistema de esta clase; para facilidad de lectura y organización se enumeran a continuación, no tienen un nivel de jerarquía, debido a que todos tienen un papel fundamental en la construcción de un curso e-learning desde el punto de vista del LMS también se puede tomar como los elementos principales que debe contener como requerimiento.

- Entorno Digital
 - Seguridad, integración, migración
- Registro y control
 - Gestión y administración, global, por curso, por usuarios
- Comunicación
 - Herramientas de comunicación sincrónica y asincrónica
- Contenidos actividades y Evaluación
 - Estándares (AICC, SCORM)

3 SCROM (Sharable Content Object Reference Model). Conjunto de especificaciones para desarrollo, empaquetamiento y distribución de material educativo en cualquier momento y en cualquier lugar.

4 ISO/IEC 9126: Estándar internacional para la evaluación del Software

- Interfase gráfica

Por otra parte debe atender a las necesidades de la institución:

- Permitir la gestión de usuarios
- Permitir la gestión de contenido (Objetos de aprendizaje, estándares de contenido).
- Tener funcionalidades de evaluación (varios tipos de preguntas, autoevaluación).
- Proporcionar un seguimiento académico para estudiantes y docentes.
- Contar con herramientas de comunicación (email, chats, foros, anuncios, etc.)
- Contar con un sistema de administración (Configuración de la plataforma)
- Proporcionar la información que apoye la toma de decisiones a nivel académico, administrativo, financiero, infraestructura. (Indicadores de gestión, servicio, disponibilidad)
- Permitir generar reportes (usuarios, cursos, evaluaciones, etc.)
- Cumplir con estándares internacionales.
- Permitir integrarse con otros sistemas (Sistemas de información académica)
- Tener una interfaz amigable, o de fácil uso.
- Considerar el crecimiento de usuarios potenciales. (Número de estudiantes de la Universidad + Número de estudiantes externos)

Como el LMS es un Software debe revisarse desde el punto de vista evaluativo obedeciendo a los aspectos técnicos de una manera no tan rigurosa de la norma ISO/IEC 9126, que busca poder medir la calidad de un programa informático.

Entendiendo por calidad “la propiedad o conjunto de propiedades inherentes al software que permiten determinar su valor”. Para ello propone la descomposición de la calidad en atributos más sencillos y fáciles de medir. De esta forma se establecen los requisitos de la calidad de un programa y se consigue un mayor conocimiento del programa estudiado.

Estos atributos son:

- Funcionalidad
- Fiabilidad
- Facilidad de Uso
- Eficiencia
- Mantenimiento
- Movilidad

2.1.1.1.-Funcionalidad

Se debe apreciar un grado de satisfacción en cuanto a las funciones que se requiere y las que están disponibles por parte de los usuarios que manipulan el sistema, independientemente del papel q desempeñen. En cuanto a uso del software, este cumplirá con los requerimientos de los profesores de la institución, además de contar con la posibilidad de agregar componentes y complementos (plugins) para garantizar la creación de objetos de aprendizaje que alcancen los objetivos.

2.1.1.2.-Fiabilidad

El sistema será fiable si garantiza la operatividad del sistema (Ej. Donde se evidencia carencias es en las caídas de red pero es un agente externo al software). El software contará con foros y wikis que darán ayuda a los

usuarios en la solución de problemas y dudas presentando una mejora continua enriquecida por administradores, desarrolladores y profesores.

2.1.1.3.-Facilidad de Uso

Las secuencias de uso, manipulación, creación y desarrollo se pueden tomar de alguna manera intuitiva y otras de manera deductiva, que se refuerzan con la interacción frecuente con la plataforma. La plataforma será fácil de comprender al generar niveles de aprehensión favorables, tendiendo a optimizar los procesos operativos de los usuarios y sus roles.

2.1.1.4.-Eficiencia

El software será eficiente al alcanzar los objetivos de aprendizaje propuestos, de esta manera la eficiencia se apreciará en los indicadores de gestión. Los tiempos de respuesta dependen de la velocidad de conexión, los recursos que brinda son necesarios y satisfactorios en su gestión.

2.1.1.5.-Mantenimiento

El software es fácil de modificar si garantiza la facilidad de extensiones, modificaciones, correcciones, así como la facilidad de poder realizar backups de la plataforma, de los cursos y realizar pruebas sobre la plataforma.

2.1.1.6.-Portabilidad

Siendo un sistema basado en normas SCROM, tendrá la facilidad para ser transferido desde una plataforma a otra (capacidad de instalación, reemplazamiento, adaptabilidad y co-existencia).

2.2.- SELECCIÓN DEL LMS

Para la selección del LMS vamos a tomar como referencia la metodología de evaluación de JOIN la cual se basa en un 'Lista de criterios' que se traducen en un 'Cuestionario de evaluación'⁵ compuesto por 120 preguntas. Dicho cuestionario está destinado al equipo de evaluación de JOIN y al equipo de desarrollo del producto. La información contenida en este cuestionario, una vez completado, se reelabora en forma de tabla y es lo que se conoce como 'Ficha técnica' del producto que acompaña la 'Ficha del producto' de carácter más informativo y generalista, formando así el paquete informativo del LMS.

En la actualidad existe una gran variedad de LMS, por esta razón para la preselección de LMS se tuvo q escoger a uno de las 3 tendencias más representativas en el panorama actual, obteniendo así:

- **Dokeos.** De la tendencia de entornos centrados en la administración de contenidos de cursos y colaboración en la modalidad de auto-estudio.
- **Moodle.** De la tendencia de entornos centrados en la comunicación y las actividades de enseñanza-aprendizaje que incluyen, también, herramientas para gestionar materiales.
- **ATutor.** De la tendencia de entornos centrados en la creación, gestión y distribución de contenidos, con algunas herramientas de comunicación añadidas, pero en segundo plano.

2.2.1.- COMPARACIÓN DE LOS LMS

En esta sección presentaremos los paquetes informativos de producto, para las plataformas preseleccionadas de acuerdo a la última evaluación de JOIN. Cabe recalcar que desde dicha evaluación han existido nuevos releases de estos tres productos (mejoras continuas), por tal razón se

⁵ http://www.guidance-research.org/sigossee/join/sp/docs/metodologia/evaluacion_pdf/attach/Quantified_methodology.pdf

utilizarán para el desarrollo las últimas versiones confiables lanzadas a la red por los creadores de los productos al momento de iniciar con la realización del proyecto aplicativo.

2.2.1.1.-Producto: Dokeos

Descripción del producto. Dokeos es un entorno de e-learning y una aplicación de administración de contenidos de cursos y también una herramienta de colaboración. Es software libre y está bajo la licencia GNU GPL, el desarrollo es internacional y colaborativo. También está certificado por la OSI y puede ser usado como un sistema de gestión de contenido (CMS) para educación y educadores, lo cual lo convierte en un LCMS (Learning Content Management System).

Sitio oficial del proyecto: <http://www.dokeos.com/>

2.2.1.1.1.- *Ficha del Producto*

Tabla 2.1. Ficha de Producto Dokeos

Ficha del producto	
Nombre LMS	Dokeos
Equipo de desarrollo	El equipo está liderado por Thomas De Praetere y colaboran con él más de diez desarrolladores más, además también contribuyen algunas universidades.
Servicio de soporte	info@dokeos.com
Número de la versión evaluada	2.0
Tecnología utilizada	PHP, Apache, MySQL
Sistemas Operativos soportados	Todos los Sistemas Operativos en los que se pueda instalar PHP, Apache y MySQL (e.g. Windows, Linux, UNix, etc)
Requisitos de Bases de Datos/Software	Apache, PHP 4.x o posterior (con soporte habilitado para mysql, zlib, preg y xml) y MySQL 3.23.6 o posterior.
Tipo de licencia	GPL
Estandares de E-Learning soportados	SCORM 1.2
Lenguas disponibles	En la actualidad están disponibles más de cuarenta lenguas (árabe, croata, holandés, inglés, finlandés, francés, alemán, italiano, japonés, portugués, ruso, español y muchas más)
Características principales	Dokeos es un fork bastante reciente de Claroline. Ambas herramientas son similares, pero Dokeos muestra su propia personalidad ahora. La aproximación diferente para los caminos de aprendizaje, la compatibilidad SCORM runtime, la organización distinta de algunas herramientas,

	<p>permiten decir que Dokeos es más que una operación estética de Claroline. La nueva versión examinada también ofrece tecnología para trabajos Drag & Drop (iPad).</p>
Posibilidad de gestionar los contenidos de los cursos	<p>Se pueden crear y gestionar cursos desde dentro de Dokeos y es posible realizar contenidos (páginas) y tests.</p>
Aspectos positivos	<p>Alta facilidad de uso y fiabilidad</p> <p>Soporte a caminos de aprendizaje</p> <p>Chat, forum, Video/audio conferencing y muchas otras herramientas de comunicación</p> <p>Test y evaluación</p> <p>Módulos de importación de paquetes SCORM y soporte SCORM runtime</p> <p>Soporte multilingüe</p> <p>Alta modularidad con tecnología plug-in</p>
Aspectos a mejorar	<p>Las herramientas de autor pueden mejorar</p> <p>La documentación no está completa y existe sólo en inglés</p>
Principales instalaciones en el mundo	<p>Hay varias instalaciones en todo el mundo, la lista está en el website.</p>
Instalaciones en Italia	<p>Campus Imedia</p> <p>Istituto Bergese</p> <p>Università di Parma</p> <p>Laser Virtual Campus</p>
Notas	<p>Dokeos es una plataforma muy interesante, la comunidad de desarrollo es amplia y su Dream map, que es la lista de desarrollos deseados, es realmente apasionante.</p>
Enlace a la versión demo	<p>http://demo.dokeos.com/index.php?language=spanish</p>

Fuente: JOIN (www.ossite.org)

Elaborado por: PROAÑO, Marco

2.2.1.1.2.- Ficha Técnica

Tabla 2.2. Ficha Técnica Dokeos

Requisitos Funcionales	
Didáctica/Administración del aprendizaje	Dokeos permite crear itinerarios formativos (camino de aprendizaje) para el curso. Cada itinerario puede ser creado de uno o más módulos, cada uno de estos módulos contiene un conjunto de recursos disponibles en el curso. Una vez que los módulos son escogidos, podemos configurar el orden y las dependencias entre módulos y entre los ítems incluidos.
Pruebas y Evaluaciones	Las pruebas pueden ser creadas dentro del sistema, almacenando preguntas en un banco de preguntas. En esta versión es posible importar paquetes HotPotatoes. Están disponibles los siguientes tipos de preguntas: selección simple o múltiple, concordancia de respuestas (Matching) y completar en el blanco. Es posible realizar un seguimiento y ver las estadísticas de las pruebas. Estas estadísticas están disponibles para cada usuario.
Formato de Medios / Autoría (producción)	El contenido del curso puede ser generado dentro del sistema (visto como archivos HTML), e importado en cualquier formato (PDF, Doc, txt, HTML, etc). Los módulos SCORM también son soportados dentro de la plataforma, incluso los resultados pueden ser vistos en forma de estadísticas. En esta versión es posible exportar itinerarios de aprendizaje en formato SCORM. El contenido del curso puede ser organizado en carpetas o directorios.
Monitoreo del curso	Los usuarios pueden ser inscritos en un curso por el administrador o ellos pueden inscribirse por ellos mismos. Es posible monitorear mucha información (acceso a herramientas, documentos, accesibilidad, etc)

Comunicación	Para la comunicación sincronía están disponibles el Chat y la conferencia de audio y video, en tanto que para la comunicación asíncrona, el mail interno (baúl de tareas), foros y archivos compartidos (publicación de los estudiantes). En esta versión y gracias a la tecnología Plug-in esta disponible RSS feed
Look and feel del sistema	Dokeos hace uso de archivos CSS y es posible personalizarlo casi todo, pero estas personalizaciones son a nivel de la plataforma (los usuarios no pueden escoger un diferente estilo o cambiarlo este archivo)
Implementación de Permisos	Dokeos hace uso de archivos CSS y es posible personalizarlo casi todo, pero estas personalizaciones son a nivel de la plataforma (los usuarios no pueden escoger un diferente estilo o cambiarlo este archivo)
Otras Características	Mas de 30 lenguajes son soportados por Dokeos (Brazilian, Croatian, Dutch, English, Esperanto, French, German, Italian, Persian, Portuguese, Serbian, Slovenian, Spanish, Swedish, y mucho más). También es soportado el módulo de autenticación LDAP. Gracias al módulo de publicaciones de los estudiantes, los usuarios pueden compartir notas o archivos, como una forma de crear un nuevo ambiente de trabajo.
Mantenibilidad	
La calidad de documentación técnica	En la actualidad los documentos para el desarrollador están disponibles en español, inglés y francés. Este documento está disponible en sección denominada Development Guidelines del sitio Web de dokeos, que es una clase de manual que contiene información acerca de la arquitectura, estructura de la base de datos, metodología de desarrollo, convenciones de código, etc. El API esta disponible en la url: www.phpxref.com/xref/dokeos/nav.html.gz?_functions/index.html.gz .

Escalabilidad	Existen muchas instituciones y universidades que están usando Dokeos. Se estima que por cada instalación más de cinco mil usuarios y 150 cursos pueden ser soportados, pero no es claro si el sistema crece horizontalmente. El sistema es muy confiable.
Extensibilidad	El código es claro y las convenciones de código son normalmente respetadas. En estos momentos la documentación del API está completada como se enunció anteriormente, y la oportunidad de ampliar las características con plug-ins es realmente importante, actualmente existen algunos plug-ins disponibles. Los requerimientos del usuario y problemas son manejados dentro del foro de Dokeos y existe un sistema CVS público.
Adaptabilidad / Cumplimiento de estándares	El sistema Dokeos cumple con el estandar SCORM 1.2 y es posible importar y exportar contenidos SCORM. También soporta el estándar IEEE LOM pero solamente para las herramientas documentos y grupos.
Facilidad de Uso	
Documentación de Usuario	La documentación esta disponible para los usuarios para su edición en diferentes lenguas. Para el usuario existe el manual del administrador, profesor y estudiante
Generación de contenido	El sistema es realmente fácil de usar y es posible crear pruebas, contenidos, entre otros dentro de la plataforma, con la opción de modificar lo que nosotros hayamos creado.
Nivel de experticia necesaria	Todas las operaciones del sistema, en cualquier nivel, no requieren de una especial experticia para el manejo de Dokeos. Cualquier operación puede ser ejecutada en forma simple sin ningún tipo de entrenamiento.

Otras características	El sistema es fácil de navegar, el idioma puede ser seleccionado por los usuarios y las personalizaciones son posibles a nivel de plataforma. Las modificaciones gráficas serán disponibles únicamente un plug-in. Los contenidos y datos disponibles en Internet como los módulos SCORM, pueden ser usados en cualquier tipo de documento (formato) y páginas HTML. Los permisos del sistema realmente no son flexibles pero es fácil de entender.
Calidad del Soporte	
Soporte	Es posible acceder a muchos soportes para entrenamiento, consulta, hosting, etc. Para cualquier consulta o información es posible escribir a l mail: info@dokeos.com. El equipo de Dokeos está siempre atento ante cualquier requerimiento y por otro lado existe mucha información útil en el foro de Dokeos.
Coste Total	
Tipo de usuarios	Colegios y universidades son los usuarios predominantes, pero también las compañías o instituciones públicas. Un listado en el sitio Web de Dokeos incluye solamente organizaciones que están usando los servicios comerciales de Dokeos y el número de usuarios es bastante grande.
Estabilidad financiera	El proyecto inició en el 2003, y el equipo núcleo está compuesto de alrededor de 7 miembros, pero hay pocos desarrolladores alrededor del mundo. El equipo se funda por personas comprometidas por colaborar y por servicios proporcionados. Solo en los últimos 2 años se ha logrado un avance al recibir todo aporte de colaboradores/usuarios al rededor del mundo.
Coste inicial de establecimiento del sistema	No existen especiales requerimientos de hardware. Los requerimientos de software son: Apache, MySQL, PHP y algunos módulos PHP (mysql, zlib, preg, xml). La puesta en marcha del sistema toma menos de un día.

Costos recurrentes	Actividades periódicas son las mismas para todo sistema Web: parches en el sistema operativo o instalación de herramientas.
---------------------------	-----------------------------------------------------------------------------------------------------------------------------

Fuente: JOIN (www.ossite.org)

Elaborado por: PROAÑO, Marco

2.2.1.2.-Producto: Moodle

Descripción del Producto. Moodle es uno de los LMS más populares y está actualmente viviendo una fase explosiva de expansión. Su comunidad de usuarios y desarrolladores es muy numerosa y se caracteriza por su entusiasmo respecto al sistema. Moodle es un proyecto inspirado en la pedagogía del constructivismo social.

Sitio oficial del proyecto: <http://www.moodle.org>

2.2.1.2.1.- Ficha del Producto

Tabla 2.3. Ficha del Producto Moodle

Ficha del producto	
Nombre LMS	Moodle
Equipo de desarrollo	En 1999 Martin Dougiamas (Australia) inició el proyecto Moodle. Actualmente colaboran en él alrededor de 100 personas entre desarrolladores (cerca de 50), traductores (otros 40), beta-testers ...
Servicio de soporte	Desde el año 2004 en http://www.moodle.com los Moodle Partners (grupo de empresas de servicios) ofrecen un amplia oferta de servicios comerciales para usuarios, entre los cuales hosting Moodle completo, contratos de soporte remoto, desarrollos a medida y consultoría.
Número de la versión evaluada	1.5
Tecnología utilizada	PHP
Sistemas Operativos soportados	Unix, Linux, Windows, Mac OS X, Netware y cualquier otro sistema que soporte PHP.
Requisitos de Bases de Datos/Software	PHP version 4.1.0. o mayor. Cualquier servidor web que soporte PHP (la mayoría de las instalaciones usan Apache). Una base de datos, MySQL y PostgreSQL especialmente, pero también puede usarse Oracle, Access, Interbase, ODBC y otras.
Tipo de licencia	GPL
Estandares de E-Learning soportados	Importación SCORM 1.2 y IMS-QTI

Lenguas disponibles	40 lenguas incluidas Árabe, Catalán, Chino (simplificado y tradicional), Checo, Danés, Holandés, Inglés (versiones UK y US), Finlandés, Francés (versiones Francia y Canadá), Alemán, Griego, Húngaro, Indonesio, Italiano, Japonés, Noruego, Polaco, Portugués (Portugal y Brasil), Rumanés, Ruso, Slovaco, Español (versiones España, Méjico, Argentina y Caribe), Sueco, Tailandés y Turco.
Características principales	La principal característica de Moodle, aparte de su fundamento en la pedagogía del constructivismo social, es su gran y continuamente creciente comunidad de usuarios que le da al sistema una enorme vitalidad.
Posibilidad de gestionar contenidos.	Moodle permite crear y gestionar contenidos de cursos y tests.
Aspectos positivos	El sistema es muy intuitivo y fácil de usar, está traducido a más de 40 lenguas, se apoya en una gran comunidad de usuarios y desarrolladores, se tiene ya la documentación Moodle Documentation Project basado en Wiki, trabaja bajo plantillas.
Aspectos a mejorar	Se debe mejorar el desarrollo libre de plantillas, Moodle no soporta las especificaciones de accesibilidad (ej. W3C WAI), se está trabajando en la mejora continua de los módulos.
Principales instalaciones en el mundo	Miles de instalaciones por todo el mundo, listadas en http://www.moodle.org/sites
Notas	El desarrollo e implantación que Moodle está teniendo es realmente espectacular. Su gran comunidad de usuarios y desarrolladores le confieren un sello especial de calidad y continuidad al proyecto. Es una opción a tener seriamente en cuenta a la hora de escoger un LMS de software libre.
Enlace a la versión demo	http://www.moodle.org

Fuente: JOIN (www.ossite.org)

Elaborado por: PROAÑO, Marco

2.2.1.2.2.- Ficha del Técnica

Tabla 2.4. Ficha Técnica Moodle

Requisitos Funcionales	
Didáctica/Administración del aprendizaje	El sistema ofrece algunos esquemas de cursos pre-estructurados. Los tutores no pueden definir itinerarios formativos condicionales o alternativos. El sistema no soporta planes de aprendizaje. El sistema parcialmente soporta el portafolio del estudiante (es posible seguir una secuencia de todas la actividades del estudiante dentro de un curso, pero el concepto global de portafolio del estudiante no existe). El estudiante recibe retroalimentación de diferentes maneras dependiendo del tipo de actividad que el estudiante este realizando. En cualquier momento el estudiante puede ver el estado de su trabajo y de muchas actividades, el sistema envía emails de alerta para informar al estudiante acerca de sus principales problemas.
Pruebas y Evaluaciones	El sistema ofrece la funcionalidad de Autoría (producción) para la generación y modificación de pruebas. Las pruebas pueden ser importadas de varios formatos. Diferentes tipos de preguntas puede ser empleados para las pruebas: selección simple o múltiple, concordancia de respuestas de términos y figuras (Matching), entre otras. No hay funcionalidad de servicios. El sistema no puede generar certificados después de finalizar una carga de trabajo pero un módulo de certificados está siendo desarrollado.

<p>Formato de Medios / Autoría (producción)</p>	<p>El contenido puede ser creado dentro del sistema. Cualquier contenido Web es soportado y puede ser importado. Los cursos de Moodle y los paquetes SCORM 1.2 pueden ser importados. También los cursos pueden ser exportados (en formato propietario XML) con asociados metadatos. Los datos de los usuarios pueden ser importados o exportados desde Moodle (formato XML, texto, DB, LDAP).</p>
<p>Monitoreo del curso</p>	<p>En el sistema, los cursos pueden ser programados (período de registro o bloqueo de cursos). Los estudiantes pueden inscribirse por si mismo en un curso o los instructores ratifican la inscripción. Existe la funcionalidad para el seguimiento de los usuarios: los datos son presentados en tablas y pueden ser exportados. Los datos que pueden ser monitoreados son: resultados de pruebas, accesibilidad a la plataforma del usuario.</p>
<p>Comunicación</p>	<p>La comunicación síncrona es soportado por Moodle a través del Chat y la comunicación asíncrona a través del e-mail y foro. Herramientas externas para comunicación (pizarras, salas de chat) pueden ser integrados al sistema como programas externos. El sistema soporta RSS/blogging.</p>
<p>Look and feel del sistema</p>	<p>El diseño del sistema puede ser personalizado en pequeños aspectos. Los usuarios no pueden personalizar el diseño del sistema. Es posible personalizar la página de login.</p>
<p>Implementación de Permisos</p>	<p>Los permisos son manejados a través de la implementación de roles. Los roles identificados son: administradores, profesores, tutores, estudiantes e invitados. Los usuarios pueden ser organizados en grupos muy fácilmente.</p>

Otras Características	El sistema hace uso de la encriptación en la fase de login. No existe alguna funcionalidad para manejar pagos pero el sistema soporta Paypal. Los usuarios pueden registrar el contenido de aprendizaje pero no pueden compartir anotaciones con otros usuarios. Más de 40 idiomas son soportados. El sistema provee de mecanismos de autenticación LDAP.
Mantenibilidad	
La calidad de documentación técnica	Existe documentación para los desarrolladores en más de 4 idiomas. La calidad de la documentación es buena pero podría ser más explícita. Actualmente se está trabajando sobre la documentación del código con phpdoc.
Escalabilidad	El máximo número de usuarios activos de una instalación es de alrededor de 5000 usuarios, de igual manera, el número máximo de cursos dentro una instalación es de alrededor 150 cursos. Es posible administrar diferentes clientes en una instalación. El sistema puede crecer horizontalmente (agrupación de webserver con sesiones almacenadas en la base de datos). El sistema es muy confiable
Extensibilidad	Existen mecanismos documentados para la extensión de características en el proceso de desarrollo, en este sentido existe el concepto de plug-in. Existe parcialmente disponible una API definida. El sistema es desarrollado (en partes) con una aproximación orientada a objetos. Los desarrolladores se rigen a ciertas convenciones de código definidas para el LMS. La calidad del código fuente del proyecto es buena. No existe una metodología para la gestión de requerimientos de usuario. Existe un sistema de reporte de errores o fallas empleado por el

	equipo de desarrollo. El sistema ha tenido 6 releases de importancia y 15 releases de menor importancia.
Adaptabilidad / Cumplimiento de estándares	El sistema es modularizado en partes. Moodle cumple con el estándar SCORM 1.2 (reproducción de paquetes) y el estándar IMSQTI (importación).
Facilidad de Uso	
Documentación de Usuario	Existe ayuda en línea en el sistema y además la plataforma cuenta con documentación de calidad en varios idiomas (mas de 40) para usuarios, profesores y administradores.
Generación de contenido	El contenido de aprendizaje puede ser generado y modificado empleado las funcionalidades del sistema pero esto toma algún tiempo. Las pruebas pueden ser creadas y modificadas dentro del sistema.
Nivel de experticia necesaria	Poca experticia es requerida para usar la plataforma como un tutor o usuario. Para los administradores y profesores más experticia es necesaria debido a que el sistema es complejo y rico en características.
Otras características	Algún material de entrenamiento, además de la ayuda normal puede ser obtenida de manera libre de http://www.moodle.org . El sistema es fácil de manejar. El diseño del sistema puede ser configurado para cualquier organización. Los permisos del sistema son simples y muy fáciles de entender. El sistema no cumple con ninguna especificación de accesibilidad.

Calidad del Soporte	
Soporte	Las instituciones pueden adquirir algún soporte comercial en Moodle en http://www.moodle.com sobre las bases del contrato y garantía. Moodle provee de conjunto de servicios comerciales para los usuarios, incluyendo el servicio de hosting, contratos de apoyo remoto, desarrollo de código personalizado y consulta. El equipo de Moodle está atento para brindar soporte. Una respuesta a una inquietud en el foro toma unas pocas horas.
Coste Total	
Tipo de usuarios	Los usuarios predominantes son las escuelas (25%), pequeñas y medianas empresas (25%), universidades (15%) y la industria (15%). Actualmente hay más de 2000 sitios registrados usando Moodle. Los clientes puede ser vistos en la dirección: http://www.moodle.org/sites
Estabilidad financiera	El proyecto inició en 1999 con el desarrollo de Martin Dougiamas. Actualmente el equipo de desarrollo tiene cerca de 100 miembros (desarrolladores, traductores, testers, etc.).
Coste inicial de establecimiento del sistema	La memoria y el disco necesaria depende del número de usuarios, nivel de concurrencia en el sistema y principalmente la características de los cursos. Los requerimientos de software son un servidor Web, una base de datos (principalmente MySQL) y PHP con algunas librerías opensource compiladas, que están incluidas en la distribución de Moodle. El sistema también necesita hardware y software adicional para copias de seguridad, conectividad, etc. Moodle puede ser configurado en menos de 30 minutos. Dependiendo sobre los enlaces a sistemas externos (LDAP, otra base de datos) y otros

	factores, la instalación de Moodle debería operar sin problemas.
Costos recurrentes	Horas de entrenamiento recomendado para un administrador 20-30 h., para un profesor: 50-100 h., para un tutor: 20-30 h., para un usuario: 5-10 h.

Fuente: JOIN (www.ossite.org)

Elaborado por: PROAÑO, Marco

2.2.1.3.-Producto: Atutor

Descripción del producto. Este proyecto empezó en 2002 en colaboración con el Adaptive Technology Resource Centre (ATRC) de la Toronto University. Este centro es un líder internacionalmente reconocido en el desarrollo de tecnologías y estándares que permitan a la gente con discapacidades el acceso a las oportunidades e-learning y esta misión ha influenciado profundamente el desarrollo de la plataforma. El desarrollo ha prestado especial interés a la accesibilidad: ATutor es la única plataforma LMS que cumple las especificaciones de accesibilidad W3C WCAG 1.0 de nivel AA+.

Sitio oficial del proyecto: <http://www.atutor.ca>

2.2.1.3.1.- Ficha Del Producto

Tabla 2.5. Ficha del producto Atutor

Ficha del producto	
Nombre LMS	Atutor
Equipo de desarrollo	Greg Gay (lider del proyecto) y unos cuantos desarrolladores (la lista completa está en el website) en colaboración con el Adaptive Technology Resource Centre (Toronto University).
Servicio de soporte	Es posible entrar en contacto con el equipo de desarrollo y soporte rellenando los campos de la página Contact Atutor (http://www.atutor.ca/contact.php?subject=ATutor+Service+Request)
Número de la versión evaluada	1.4.2
Tecnología utilizada	PHP, Apache, MySQL
Sistemas Operativos soportados	Todos los sistemas operativos en los cuales es posible instalar PHP, Apache y My SQL (ej. Windows, Linux, UNix, etc)
Requisitos de Bases de Datos/Software	Apache 1.2.x, PHP 4.2 o superior (con soporte habilitado para zlib y mysql) y MySQL 3.23.x o superior.
Tipo de licencia	GPL
Estandares de E-Learning soportados	SCORM 1.2, IMS packaging y las especificaciones de accesibilidad W3C WCAG 1.0 de nivel AA+
Lenguas disponibles	Más de dieciséis (danés, holandés, francés, alemán, griego, italiano, portugués, ruso, español y muchos más)
Características principales	Soporte de estándares, facilidades de importación/exportación de contenidos y usuarios, herramientas de seguimiento.

Posibilidad de gestionar los contenidos de los cursos	Se pueden gestionar cursos, es fácil crear contenidos y tests dentro de la plataforma e importar paquetes SCORM 1.2 e IMS
Aspectos positivos	Entorno integrado de autor para crear y gestionar cursos
	Soporte de palabras clave para ayudar al motor de búsqueda
	Búsqueda posible para todos los cursos del catálogo
	Herramientas de importación/exportación de usuarios
	Soporte de paquetes SCORM y IMS
	Herramientas de accesibilidad
	Disponibilidad de herramientas de test y evaluación
Aspectos a mejorar	SCORM runtime no soportado
	Documentación sólo en inglés
	Faltan herramientas síncronas
Principales instalaciones en el mundo	Hay varias instalaciones por todo el mundo: la lista completa está en el website http://www.atutor.ca
Notas	Una plataforma muy bonita que permite hacer muchas cosas de un modo simple y donde la facilidad de uso, el perfil educacional y el aprendizaje humano son particularmente tenidos en cuenta
Enlace a la versión demo	http://www.atutor.ca/atutor/demo.php

Fuente: JOIN (www.ossite.org)

Elaborado por: PROAÑO, Marco

2.2.1.3.2.- Ficha Técnica

Tabla 2.6. Ficha Técnica Atutor

Requisitos Funcionales	
Didáctica/Administración del aprendizaje	Actualmente, los itinerarios formativos nos son disponibles y no es posible implementar actividades de aprendizaje.
Pruebas y Evaluaciones	Las herramientas son realmente fáciles de usar y las páginas HTML pueden ser creadas sin ninguna experticia. Contenidos como los paquetes SCORM o IMS pueden ser importados, además páginas HTML o cualquier formato de archivos pueden ser importados y organizados dentro de la plataforma. Las pruebas (selección múltiple, verdadero/falso, preguntas abiertas) pueden ser creadas dentro del sistema y podemos seleccionar si la calificación debe ser asignada automáticamente o por el tutor.
Formato de Medios / Autoría (producción)	Paquetes de contenidos SCORM e IMS pueden ser importados, pero también archivos de cualquier formato. Una característica importante es la posibilidad de exportar un curso entero (o parte de este) en formato del estándar ISM. Los cursos pueden ser clasificados en una simple carpeta y los contenidos pueden ser escritos con palabras claves que serán usadas para el motor de búsqueda. Los datos de los usuarios pueden ser exportados e importados en un formato CSV y todos los archivos multimedia son soportados.
Monitoreo del curso	Un curso puede ser publicado después de una fecha predefinida. Los usuarios pueden inscribirse por ellos mismos y los instructores aprueban la inscripción. Es posible monitorear toda la información (herramientas, estudiantes, etc.) y estos datos pueden ser exportados.

Comunicación	La comunicación síncrona es posible usando Chat, en tanto que la comunicación asíncrona está disponible a través de los foros, email e intercambio de archivos. Toda estas herramientas están en el sistema Acollab, un add-on para ATutor 1.4+.
Look and feel del sistema	El sistema hace uso de hojas de estilo y cada usuario puede escoger el diseño (posición) de los objetos dentro de la página, ocultando algunas cosas o simplemente incorporar uno de los disponibles plantillas. Los instructores pueden también establecer una plantilla para un conjunto de cursos.
Implementación de Permisos	Los permisos del sistema es bastante bueno y no es posible definir nuevos roles. Los roles predefinidos son: administrador, instructor, usuario e invitado, y estos son suficientes para satisfacer cualquier escenario. Es posible crear grupos y todas las herramientas de colaboración son basadas en grupos.
Otras Características	Los usuarios pueden compartir notas pero no pueden registrar un particular contenido. En estos momentos más de 16 idiomas están disponibles y es posible agregar una nota en el final de las páginas del contenido con el objeto de añadir el concepto de propiedad intelectual. Soporte LDAP no es disponible desde esta versión.
Mantenibilidad	
La calidad de documentación técnica	Existe un sitio Web donde las directrices de los desarrolladores está bien descrito (existe solamente la versión en inglés).
Escalabilidad	El sistema no es muy popular, pero existen muchos demos e instalaciones oficiales alrededor del mundo. El sistema es también confiable.

Extensibilidad	El código es bastante bueno y las convenciones de código son casi respetadas. No existe el concepto de plug-in y además el API del sistema no está definido. Los requerimientos del usuario y problemas son manejados dentro del foro. Subversion es una herramienta que ha sido adoptada para el repositorio y el versionamiento de la plataforma. Actualmente existen 6 versiones estables.
Adaptabilidad / Cumplimiento de estándares	El sistema soporta empaquetamiento IMS para importar y exportar funcionalidades y SCORM 1.2 solamente para importar. El sistema es modular en el sentido que las funciones son agrupadas.
Facilidad de Uso	
Documentación de Usuario	La documentación esta solamente disponible en inglés y para cada actor involucrado en el escenario (administradores, instructores y estudiantes). También existe ayuda en línea.
Generación de contenido	Los instructores pueden fácilmente crear y modificar contenido, incluso es posible modificar contenido SCORM dentro de un editor. De igual manera es para las pruebas
Nivel de experticia necesaria	El sistema es fácil de usar y no se requiere un alto nivel de experticia. El único inconveniente es que toma algún tiempo para ver y entender todas las funcionalidades disponibles en la plataforma.
Otras características	Los materiales a ser usados en el sistema pueden ser encontrados en formato SCORM LO, IMS o cualquier archivo con cualquier formato. El sistema es fácil de usar, personalizable y los permisos del sistema es simple de entender. Además, la plataforma cumple con las especificaciones de accesibilidad (W3C WAI).

Calidad del Soporte	
Soporte	Es posible contactar al equipo de Atutor en su sitio Web (www.atutor.ca/contact.php?subject=ATutor+Service+Request) para obtener servicios, sobre las bases de un contrato, con respecto a un nuevo desarrollo, hosting, consultas y soporte. El equipo esta siempre presto a responder cualquier inquietud.
Coste Total	
Tipo de usuarios	Los tipos predominantes de usuarios son los colegios y universidades, pero también existen pequeñas empresas y organizaciones. El número de usuarios puede ser estimado en algunos miles en todo el mundo y hay una completa lista de instituciones en su sitio Web si nosotros queremos contactarnos con ellos.
Estabilidad financiera	Este proyecto inició en el 2002 y el equipo puede ser estimado eb menos de diez personas. El equipo núcleo está consolidado y hay poco personal que brinda soporte que colabora en el proyecto.
Coste inicial de establecimiento del sistema	No existen requerimientos especiales de hardware. Los requerimientos de software son: Apache, MySQL, PHP (las versiones sugeridas se encuentran descritas en el sitio Web)
Costos recurrentes	Actividades periódicas son las mismas para todo sistema Web: parches en el sistema operativo o instalación de herramientas.

Fuente: JOIN (www.ossite.org)

Elaborado por: PROAÑO, Marco

De las descripciones técnicas podemos obtener que **Dokeos**, **Moodle** y **Atutor** no son los únicos LMS con licencia Open Source, pero **Dokeos** y **Moodle** son probablemente los más difundidos.

La comparación de Moodle y Dokeos puede dar una idea de la variedad de enfoques que los LMS pueden tener.

Moodle basa su modelo pedagógico en el constructivismo social, esto es, en el establecimiento de comunidades alrededor de un tema que realizan actividades, reflexión crítica, etc. Esto marca profundamente su organización e interfaz, construida alrededor de 3 modelos de interacción on-line:

- Weekly, en la que toda la interfaz gira alrededor de la asignación de actividades semanales.
- Topics, en la que queda organizada en base a los temas propuestos en el curso.
- Social, en la que el eje central del curso pasa a ser un foro de discusión.

Por otra parte, el modelo de Dokeos es algo distinto. La interfaz se organiza en base al concepto de curso como agrupación de distintos tipos de recursos: contenido, foro, auto-evaluaciones, etc.

Y aunque las funcionalidades son casi las mismas en ambos sistemas, dependiendo del estilo pedagógico del curso será más fácil impartirlo usando una plataforma y otra. Simplificando mucho podríamos decir que:

- En Dokeos se puede poner en marcha un curso en modo auto-estudio con elementos de colaboración y comunicación, mientras q Moodle se adapta mejor a los cursos basados en la interacción entre los participantes.
- Las herramientas ofrecidas por Moodle son mejores que las herramientas ofrecidas por Dokeos ya que abarcan mas campos de estudio y en la actualidad existen un sin fin de herramientas (plug-ins) que se pueden instalar en Moodle.

- Dokeos tiene características de LCMS (Learning Content Management System) con propiedades de LMS (Learning Management System), Moodle es simplemente LMS. Al contar ya la institución con un CMS (Content Management System)⁶ como lo es Joomla⁷ la inclinación va hacia Moodle.
- Moodle no ofrece documentación sobre su base de datos, mientras que Dokeos Ofrece una basta información sobre ella.
- Moodle posee un entorno similar de programación a Joomla (sistema bajo el que trabaja el sistema Web de la institución), también cabe mencionar su compatibilidad es casi absoluta.

A continuación se muestra una tabla comparativa que resume lo expuesto en las páginas anteriores.

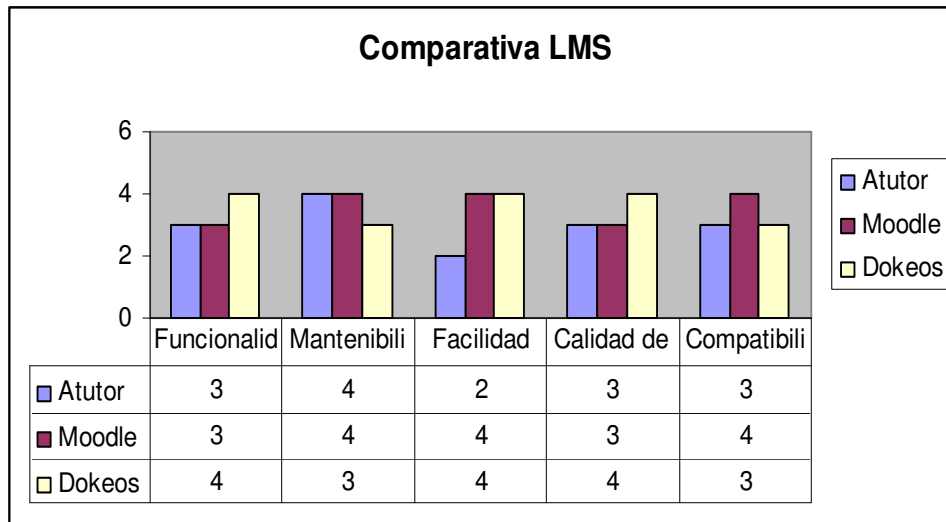
Se valora a cada apartado con un rango de 1 a 4 de acuerdo a los requerimientos de la institución, a sus disponibilidades y compatibilidades tecnológicas, que se corresponden con la siguiente valoración:

- 1. Muy deficiente
- 2. Deficiente
- 3. Bien
- 4. Excelente

⁶ www.opensourcecms.com/

⁷ www.joomlaspanish.org

Gráfico 2.1. Comparación cuantitativa de LMS

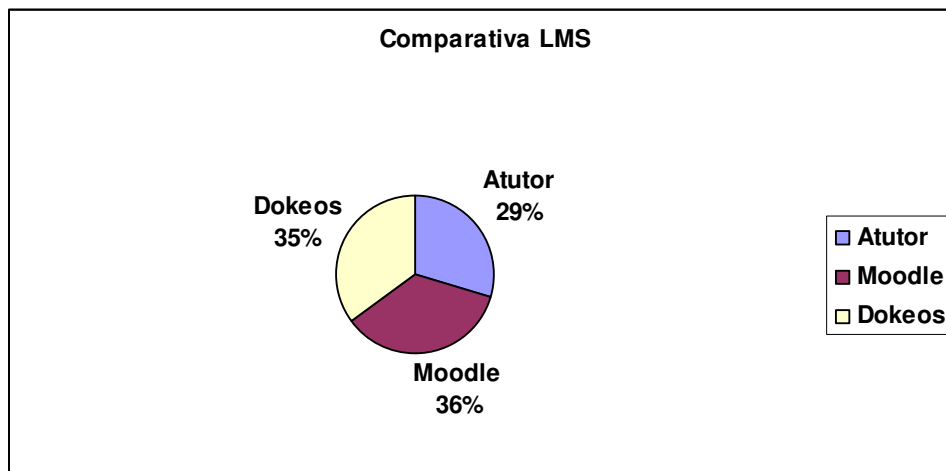


Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

Se adjunta también un gráfico que refleja a la vista los porcentajes de comparación de los LMS analizados anteriormente.

Gráfico 2.2. Comparación de LMS



Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

Debido a lo señalado anteriormente podemos determinar q Moodle ahora en su versión 1.9.x es la plataforma seleccionada por la adaptación y fiabilidad en relación a la tecnología utilizada en la escuela.

2.3.- ANÁLISIS DE LA ARQUITECTURA Y METODOLOGÍA DEL LMS SELECCIONADO

2.3.1.- DESCRIPCIÓN DE MOODLE 1.9.x

Moodle es un paquete de *software* diseñado para ayudar al profesor a crear fácilmente cursos en línea. Estos sistemas e-learning también se llaman Sistemas de Gestión de Aprendizaje (Learning Management System, LMS) o Ambientes Virtuales de Aprendizaje (AVA)⁸. Se distribuye como *software* libre bajo las normas de licencia pública (Global Public license, GPL)⁹. Básicamente, esto significa que los usuarios de Moodle tienen algunas libertades: pueden copiar, usar y modificar Moodle siempre que acepten proporcionar el código fuente a otros, no modificar o eliminar la licencia original y los derechos de autor, y aplicar esta misma licencia a cualquier trabajo derivado de él.

Moodle significa *Modular Object Oriented Dynamic Learning Environment* (Entorno Modular de Aprendizaje Dinámico Orientado a Objetos). Este sistema permite una fácil interacción entre los profesores y sus alumnos, así como entre los mismos alumnos.

2.3.2.- CRITERIOS PARA SU ARQUITECTURA

Desde la perspectiva de un administrador de sistemas, Moodle ha sido diseñado de acuerdo con los siguientes criterios:

⁸ Un Ambiente Virtual de Aprendizaje (AVA) ó Virtual learning environment (VLE) es un sistema de software diseñado para facilitar a profesores la gestión de cursos virtuales para sus estudiantes, especialmente ayudándolos en la administración y desarrollo del curso.

⁹ <http://www.gnu.org/copyleft/gpl.html>

- **Moodle se debe poder ejecutar en la más amplia posible variedad de plataformas.**

El sistema operativo de las aplicaciones Web que funcionan en la mayoría de las plataformas es PHP (Hypertext Preprocessor) combinada con MySQL, y este es el entorno en el que Moodle ha sido desarrollado (sobre Linux, Windows, y Mac OS X). Moodle también usa la librería ADOdb (Remote Data Access) para la consulta de bases de datos, lo que significa que puede usar más de diez marcas diferentes de bases de datos (desafortunadamente, a pesar de ello, no puede aún crear tablas en todas esas bases de datos).

- **Moodle debe ser fácil de instalar, aprender y modificar**

Los primeros prototipos de Moodle (1999) se construyeron usando Zope, un avanzado servidor de aplicaciones Web orientado a objetos.

Desafortunadamente, aunque la tecnología era bastante buena, tenía una curva de aprendizaje muy elevada y no era muy flexible en términos de administración del sistema. El lenguaje PHP, por otro lado, es muy fácil de aprender (especialmente si se ha hecho algo de programación usando cualquier otro lenguaje de script). Pronto se tomó la decisión de evitar usar un diseño orientado a clases, con la finalidad, una vez más, de mantenerlo fácil de entender para los principiantes. La reutilización del código se archiva en librerías con funciones claramente tituladas y con una disposición de los archivos de script, consistente. PHP es también fácil de instalar (existen versiones ejecutables para todas las plataformas) y está ampliamente disponible, pues la mayoría de los servicios de alojamiento lo proporcionan como un estándar.

- **Debe ser fácil de actualizar desde una versión a la siguiente**

Moodle sabe cuál es su versión (así como las versiones de todos los módulos) y se ha construido un mecanismo interno para que Moodle pueda actualizarse a sí mismo de forma apropiada a las nuevas versiones (por ejemplo, puede renombrar las tablas de las bases de datos o añadir nuevos campos). Usando CVS (Control Versions System) en Unix, por ejemplo, uno tan sólo tiene que introducir la instrucción "cvs update -d" y luego visitar la página principal del sitio para completar la actualización.

- **Debe ser modular para permitir el crecimiento**

Moodle tiene una serie de características modulares, incluyendo temas, actividades, interfaces de idioma, esquemas de base de datos y formatos de cursos. Esto le permite a cualquiera añadir características al código básico principal, o incluso distribuir las por separado.

- **Debe poder usarse junto a otros sistemas**

Una de las cosas que hace Moodle es mantener todos los archivos para un curso en un único directorio en el servidor. Esto podría permitir que el administrador de un sistema proporcione similares formas de acceso a un nivel de archivo para cada profesor, tal como Appletalk, SMB, NFS (Network File System), FTP (File Transport Protocol), WebDAV (Web-Based Distributed Authoring and Versioning) y demás. Los módulos de autenticación le permiten a Moodle usar LDAP (List and Data Professionals Ltd.), IMAP (Internet Message Access Protocol), POP3, NNTP (Network News Transport Protocol) y otras bases de datos como fuentes de

información de los usuarios. Por otra parte, aún queda trabajo por hacer sobre esto. Para futuras versiones de Moodle se tienen planeadas las siguientes características: importación y exportación de los datos de Moodle utilizando formatos basados en el Lenguaje de Marcado Ampliable o Extensible, XML (incluyendo IMS y SCORM), incrementar el uso de hojas de estilo para el formateo de interfaces (de manera que puedan integrarse visualmente en otros sitios Web).

2.4.- CARACTERÍSTICAS DE MOODLE

2.4.1.- TÉCNICAS

- Se ejecuta sin modificaciones bajo Unix, Linux, Windows, Mac OS X, Netware y otros sistemas operativos que permitan el protocolo PHP (la mayor parte proveedores de alojamiento Web lo permiten).
- Está diseñando de manera modular, y permite una gran flexibilidad para agregar (y quitar) funcionalidades en muchos niveles.
- Se actualiza muy fácilmente desde una versión anterior a la siguiente, puesto que conserva la misma estructura en la base de datos. Tiene un sistema interno para actualizar y reparar su base de datos cada cierto tiempo.
- Usa solamente una base de datos (si lo necesita puede compartirla con otras aplicaciones).
- Utiliza una completa consulta de bases de datos, soporta las principales marcas de bases de datos como MySQL y PostgreSQL.

2.4.2.- DISEÑO GENERAL

- Apropiaada para el 100% de las clases en línea, así como también para complementar el aprendizaje presencial.

- Tiene una interfaz de navegador de tecnología sencilla, ligera, eficiente, y compatible.
- Es fácil de instalar en casi cualquier plataforma que soporte PHP. Sólo requiere que exista una base de datos (y la pueda compartir).
- Con su completa consulta de bases de datos, soporta las principales marcas de bases de datos (excepto en la definición inicial de las tablas).
- La lista de cursos muestra descripciones de cada uno de los cursos que hay en el servidor, incluyendo la posibilidad de acceder como invitado.
- La mayoría de las áreas de introducción de texto (recursos, mensajes de los foros etc.) pueden ser editadas usando HTML, tan sencillo como cualquier editor de texto de Windows.

2.4.3.- ADMINISTRACIÓN DEL SITIO

- El sitio es administrado por un usuario administrador, definido durante la instalación. Se pueden asignar categorías de administración como creadores de curso, matriculadores y administradores equivalentes.
- Los temas permiten al administrador personalizar los colores del sitio, fuentes, presentación, etc., para ajustarse a las necesidades de una institución o particulares.
- Pueden añadirse fácilmente nuevos módulos de actividades a los ya instalados.
- Los paquetes de idiomas permiten una localización completa de cualquier idioma. Estos paquetes pueden editarse usando un editor integrado. Actualmente hay paquetes para 43 idiomas distintos.

2.4.4.- ADMINISTRACIÓN DE USUARIOS

El objetivo principal es la reducción al mínimo de la carga operativa del administrador, pero manteniendo una alta seguridad:

- Soporta un rango de módulos de autenticación, que permiten una integración sencilla con los sistemas existentes.
- Los estudiantes pueden crear sus propias cuentas de acceso. La dirección de correo electrónico se verifica mediante confirmación. Las cuentas de acceso pueden verificarse desde un servidor LDAP (Lightweight Directory Access Protocol), aunque se pueden usar otros datos de contacto del usuario como, nombres, apellidos, correo, etc.
- El administrador puede especificar qué campos usar para su identificación como IMAP, POP3, NNTP: las cuentas de acceso se verifican contra un servidor de correo o de noticias (news). Soporta los certificados cifrados SSL (Secure Sockets Layer) y TLS (Transport Layer Security).
- Cualquier base de datos que contenga al menos dos campos puede usarse como fuente externa de autenticación.
- Cada persona necesita sólo una cuenta para todo el servidor. Por otra parte, cada cuenta puede tener diferentes tipos de acceso.
- Una cuenta de administrador controla la creación de cursos y determina los profesores, asignando usuarios a los cursos.
- Una cuenta como autor de curso permite sólo crear cursos y enseñar en ellos.
- Los profesores pueden tener privilegios de edición restringidos para que no puedan modificar el curso (p.e. para tutores a tiempo parcial)
- Seguridad: los profesores pueden añadir una "clave de acceso" para sus cursos, con el fin de impedir el acceso de quienes no sean

sus estudiantes. Pueden transmitir esta clave personalmente o a través del correo electrónico personal, etc.

- Los profesores pueden inscribir a los alumnos manualmente si lo desean.
- Los profesores pueden dar de baja a los estudiantes manualmente si lo desean, aunque también existe una forma automática de dar de baja a los estudiantes que permanezcan inactivos durante un determinado período de tiempo (establecido por el administrador).
- Se anima a los estudiantes a crear un perfil en línea de sí mismos, incluyendo fotos, descripción, etc. De ser necesario, pueden esconderse las direcciones de correo electrónico.
- Cada usuario puede especificar su propia zona horaria, y todas las fechas marcadas en Moodle se traducirán a esa zona horaria (las fechas de escritura de mensajes, de entrega de tareas, etc.).
- Cada usuario puede elegir el idioma que se usará en la interfaz de Moodle (inglés, francés, alemán, español, portugués, etc.).

2.4.5.- ADMINISTRACIÓN DE CURSOS

- Un profesor sin restricciones tiene control total sobre todas las opciones de un curso, incluido el restringir a otros profesores.
- Se puede elegir entre varios formatos de curso tales como semanal, por temas o el formato social basado en debates.
- Ofrece una serie flexible de actividades para los cursos: foros, glosarios, cuestionarios, recursos, consultas, encuestas, tareas, chat y talleres.
- En la página principal del curso se pueden presentar los cambios ocurridos desde la última vez que el usuario entró en el curso, lo que ayuda a crear una sensación de comunidad.
- La mayoría de las áreas para introducir texto (recursos, envío de mensajes a un foro etc.) pueden editarse usando un editor HTML WYSIWYG integrado.

- Todas las calificaciones para los foros, cuestionarios y tareas pueden verse en una única página (y descargarse como un archivo con formato de hoja de cálculo).
- Registro y seguimiento completo de los accesos del usuario. Se dispone de informes de actividad de cada estudiante, con gráficos y detalles sobre su paso por cada módulo (último acceso, número de veces que lo ha leído) así como también de una detallada "historia" de la participación de cada estudiante, incluyendo mensajes enviados, entradas en el glosario, etc. en una sola página.
- Integración del correo. Pueden enviarse por correo electrónico copias de los mensajes enviados a un foro, los comentarios de los profesores, etc. en formato HTML o de texto.
- Escalas de calificación personalizadas. Los profesores pueden definir sus propias escalas para calificar foros, tareas y glosarios.
- Los cursos se pueden empaquetar en un único archivo zip utilizando la función de copia de seguridad. Éstos pueden ser restaurados en cualquier servidor Moodle.

2.5.- IDENTIFICACIÓN Y DESCRIPCIÓN DE LOS MÓDULOS EXISTENTES

Los módulos de la plataforma Moodle constituyen una de las herramientas de más poder que posee sobre otras plataformas porque al ser constantemente actualizados y con la aparición permanente de nuevos y con mejores características le permiten al usuario estar siempre al día. Estos módulos trabajan a manera de complementos (plugins), solo basta con ingresar a la página de Moodle.org Modules & plugins descargar el complemento necesario e instalarlo en la carpeta /mod de la plataforma.

2.5.1.- MÓDULO DE TAREAS

- Puede especificarse la fecha final de entrega de una tarea y la calificación máxima que se le podrá asignar.
- Los estudiantes pueden subir sus tareas (en cualquier formato de archivo) al servidor. Se registra la fecha en que se han subido.
- Se permite enviar tareas fuera de tiempo, pero el profesor puede ver claramente el tiempo de retraso.
- Para cada tarea en particular puede evaluarse al grupo completo de estudiantes (calificaciones y comentarios) en una única página con un único formulario.
- Las observaciones del profesor se adjuntan a la página de la tarea de cada estudiante y se le envía un mensaje de notificación.
- El profesor tiene la posibilidad de permitir el reenvío de una tarea tras su calificación (para volverla a calificar).

2.5.2.- MÓDULO DE CHAT

- Permite una interacción fluida mediante texto síncrono.
- Incluye las fotos de los perfiles en la ventana de chat.
- Soporta direcciones URL, Uniform Resource Locator, iconos gestuales, integración de HTML, imágenes, etc.
- Todas las sesiones quedan registradas para verlas posteriormente, y pueden ponerse a disposición de los estudiantes.

2.5.3.- MÓDULO DE CONSULTA

- Es como una votación. Puede usarse para votar sobre algo o para recibir una respuesta de cada estudiante (por ejemplo, para pedir su consentimiento para algo).
- El profesor puede ver una tabla que presenta de forma inmediata la información sobre quién ha elegido qué.

- Se puede permitir que los estudiantes vean un gráfico actualizado de los resultados.

2.5.4.- MÓDULO DE FORO

- Hay diferentes tipos de foros disponibles: exclusivos para los profesores, de noticias del curso y abiertos a todos.
- Todos los mensajes llevan adjunta la foto del proponente del foro.
- Las discusiones pueden verse anidadas, por rama, o presentar los mensajes más antiguos o el más nuevo primero.
- El profesor puede obligar la suscripción de todos a un foro o permitir que cada persona elija a qué foros suscribirse de manera que se le envíe una copia de los mensajes por correo electrónico.
- El profesor puede elegir a los participantes del foro, restringiendo la participación de los demás como observadores (por ejemplo, para crear un foro dedicado a anuncios).
- El profesor puede mover fácilmente los temas de discusión entre distintos foros.
- Las imágenes adjuntas se muestran dentro de los mensajes.
- Si el foro se utiliza para obtener una calificación, pueden restringirse a un rango de tiempo.

2.5.5.- MÓDULO CUESTIONARIO

- Los profesores pueden definir una base de datos de preguntas que podrán ser reutilizadas en diferentes cuestionarios.
- Las preguntas pueden ser almacenadas en categorías de fácil acceso, y estas categorías pueden ser "publicadas" para hacerlas accesibles desde cualquier curso del sitio.
- Los cuestionarios se califican automáticamente y pueden ser recalificados si se modifican las preguntas.

- Los cuestionarios pueden tener un límite de tiempo a partir del cual no estarán disponibles.
- El profesor puede determinar si los cuestionarios pueden ser resueltos varias veces y si se mostrarán o no las respuestas correctas y los comentarios.
- Las preguntas y las respuestas de los cuestionarios pueden ser barajadas (aleatoriamente) para disminuir las copias entre los alumnos.
- Las preguntas pueden crearse en HTML y con imágenes.
- Las preguntas pueden importarse desde archivos de texto externos.
- Los cuestionarios pueden responderse varias veces.
- Los intentos pueden ser acumulativos y acabados tras varias sesiones.
- Las preguntas de opción múltiple pueden definirse con única o múltiples respuestas correctas.
- Pueden crearse preguntas de respuesta corta (palabras o frases).
- Pueden crearse preguntas tipo verdadero/falso.
- Pueden crearse preguntas de emparejamiento.
- Pueden crearse preguntas aleatorias.
- Pueden crearse preguntas numéricas (con rangos permitidos).
- Pueden crearse preguntas de respuesta incrustada con respuestas dentro de pasajes de texto.
- Pueden crearse textos descriptivos y gráficos.

2.5.6.- MÓDULO DE RECURSO

- Admite la presentación de cualquier contenido digital, Word, PowerPoint, Flash, video, sonidos, etc.
- Los archivos pueden subirse y manejarse en el servidor, o pueden ser creados sobre la marcha usando formularios Web (de texto o HTML).

- Se pueden enlazar contenidos externos en Web o incluirlos perfectamente en la interfaz del curso.
- Pueden enlazarse aplicaciones Web, transfiriéndoles datos.

2.5.7.- MÓDULO DE ENCUESTA

- Se proporcionan encuestas ya preparadas (COLLES, ATTLS) y contrastadas como instrumentos para evaluar al ambiente de aprendizaje.
- Los informes de las encuestas están siempre disponibles, incluyendo muchos gráficos. Los datos pueden descargarse con formato de hoja de cálculo Excel o como archivo de texto CVS (“Sistema de Control de Versiones”).
- La interfaz de las encuestas impide la posibilidad de que sean respondidas solo parcialmente.
- A cada estudiante se le informa sobre sus resultados comparados con el promedio del grupo.

2.5.8.- MÓDULO DE TALLER

- Permite la evaluación de documentos entre iguales, y el profesor puede gestionar y calificar la evaluación.
- Admite un amplio rango de escalas de clasificación posibles.
- El profesor puede suministrar documentos de ejemplo a los estudiantes para practicar la evaluación.
- Es muy flexible y tiene muchas opciones.

2.6.- REQUERIMIENTOS FÍSICOS PARA PRESTAR SERVICIO EN LA ESPE-L

2.6.1.- SERVIDOR

Denominamos a éste como un dispositivo de un sistema que resuelve las peticiones de otros elementos del sistema, llamados clientes. El equipo propuesto para Moodle y todas las aplicaciones que se requieran en la institución como portal de contenidos, bases de datos, etc., implican la adquisición de un servidor dedicado con las siguientes características mínimas:

- De 1 ó 2 procesadores Intel Itanium a 1,1GHz hasta 1,6 GHz o más.
- De 4 a 6 MB de caché.
- De 1 a 2 GB de memoria DDR. Soporte hasta 16 GB.
- De 1 a 4 discos tecnología SCSI.
- Slots de expansión 64bit/133MHz y 64bit/66MHz PCI.
- Fuentes de alimentación redundantes.
- Que soporte Linux Red Hat Enterprise, Linux SuSE Enterprise Server 9, Microsoft Windows Server 2003.

2.6.2.- ESTACIONES DE TRABAJO

- Moodle trabaja en cualquier computador que tenga instalado un navegador de Internet en el que pueda correr PHP.
- Requiere una conexión mínima a Internet de 52 Kbps vía MODEM.
- Se recomienda usar computadores de última generación con 256 MRAM o más.
- Para un buen desempeño, se recomienda un sistema operativo tipo Windows XP service pack 2 o windows 2000 service pack 4. Tener

instaladas las últimas versiones de paquetes importantes como: DirectX, Java Virtual Machine, Flash player, Acrobat Reader, Quick time.

CAPÍTULO 3.

3. SELECCIÓN DE LA METODOLOGÍA Y LAS HERRAMIENTAS DE DESARROLLO

3.1.- SELECCIÓN DE LA METODOLOGÍA

En esta sección se efectuará una descripción de manera global del problema con la cual se procederá a la descripción de dos metodologías de posible selección enfocadas al desarrollo de proyectos como el nuestro.

3.1.1.- METODOLOGÍAS ÁGILES VS METODOLOGÍAS TRADICIONALES

Para la correcta selección de una metodología para el desarrollo del proyecto se debe saber diferenciar entre las metodologías Ágiles y las metodologías Tradicionales por tal motivo se realizan a continuación breves análisis de los mencionados tipos de Metodologías.

3.1.1.1.- Metodologías Tradicionales

Al inicio el desarrollo de software era artesanal en su totalidad. La ausencia de procesos formales, lineamientos claros, determinaron que se importara la concepción y fundamentos de metodologías existentes en otras áreas, y adaptarlas al desarrollo de software. Esta nueva etapa de adaptación contenía el desarrollo dividido en etapas de manera

secuencial, que de algo mejoraba la necesidad latente en el campo del software.

Entre las principales metodologías tradicionales tenemos los ya tan conocidos RUP y MSF entre otros, que centran su atención en llevar una documentación exhaustiva de todo el proyecto y centran su atención en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto.

Otra de las características importantes dentro de este enfoque, son los altos costos al implementar un cambio y la falta de flexibilidad en proyectos donde el entorno es volátil.

Las metodologías tradicionales (formales) se focalizan en documentación, planificación y procesos (plantillas, técnicas de administración, revisiones, etc.), a continuación se detalla RUP y MSF uno de los métodos más usados dentro de los métodos tradicionales.

3.1.1.2.-Metodologías Ágiles

Luego de varias opiniones tanto a favor como en contra de las metodologías tradicionales se genera un nuevo enfoque denominado, métodos ágiles, que nace como respuesta a los problemas de las metodologías tradicionales y se basa en dos aspectos puntuales, el retrasar las decisiones y la planificación adaptativa; permitiendo potenciar aún más el desarrollo de software a gran escala.

Como resultado de esta nueva teoría se crea un Manifiesto Ágil¹⁰ cuyas principales ideas son:

¹⁰ <http://www.manifiestoagile.com>

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados.
- Es más importante crear un producto software que funcione a tener que escribir documentación exhaustiva.
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos.
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Entre los principales métodos ágiles tenemos el XP (eXtreme Programming), Scrum, Iconix, Cristal Methods, AUP entre otras.

Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. Nos lo proponen porque para muchos clientes esta flexibilidad será una ventaja competitiva y porque estar preparados para el cambio significar reducir su coste.

Tabla 3.1. Comparativa Metodologías Ágiles vs. Tradicionales¹¹

METODOLOGÍA ÁGIL	METODOLOGÍA TRADICIONAL
Pocos Artefactos. El modelado es prescindible, modelos desechables	Más artefactos. El modelado es esencial mantenimiento de modelos.
Pocos roles, más genéricos y flexibles	Más roles, más específicos
No existe un contrato tradicional, debe ser bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo (además In-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones

¹¹ Maestría en Ingeniería de Sistemas con Mención en Tecnología de la Información. Lima-2007

Orientada a proyectos pequeños. Corta duración (entregables frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio.	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivos/ usados en proyectos grandes y con equipos posiblemente dispersos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos
Se esperan cambios durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

Elaborado por: PROAÑO, Marco

Con todo lo expuesto anteriormente y remitiéndonos al alcance del proyecto de Desarrollo de un prototipo de aula virtual (learning management system) utilizando Open Source para la Escuela Politécnica del Ejército Extensión Latacunga, siendo este de mediana proporción y con cambios constantes durante el desarrollo del mismo. Nos centraremos en el análisis de las metodologías: eXtreme Programming¹² (XP) y SCRUM¹³.

3.1.2.- DESCRIPCIÓN DEL PROBLEMA

En la sección 1.3 y 1.4 se realizó un análisis sobre el LMS que se seleccionó. En la sección 1.3 el análisis se basó en la arquitectura y metodología de desarrollo de Moodle en su versión 1.9, y por otro lado en la sección 1.4, se identificaron las diferentes características con las que cuenta actualmente la plataforma. Es preciso mencionar que desde la versión 1.5.x de Moodle se tienen versiones totalmente estables.

¹² Programación Extrema – Metodología de desarrollo Ágil. Metodología creada por Kent Beck.

¹³ Scrum - Metodología de desarrollo Ágil. Enfocado en la generación de valor en el mínimo tiempo.

Además, tomando en cuenta que este proyecto tiene como objetivo apoyar el proceso de enseñanza-aprendizaje en programas de pregrado y postgrado de la Escuela Superior Politécnica del Ejército Sede Latacunga. En este sentido, se ve necesario que dentro la plataforma existan módulos de gestión de carreras o programas académicos de pregrado y postgrado.

Cabe recalcar, que como herramienta o componente de un curso es necesario la incorporación de un módulo de publicación de calificaciones en diferentes categorías (pruebas, exámenes, exposiciones, etc.) por parte del profesor.

Auspiciante, cliente o interesado:

- Escuela Politécnica del Ejército Sede Latacunga
- Integrantes del grupo de tesis (Director, Codirector, Graduando, Dpto. TIC's ESPE-L)
- Los requerimientos del proyecto están definidos por el grupo de trabajo.

Recursos:

- Humano: Integrantes del grupo de tesis, dos personas encargadas de las revisiones (Director, Codirector) y dos personas encargada del desarrollo (Graduando, Delegado de las TIC's).
- Económico: Limitado.
- Tecnológico:
 - Hardware: Se cuenta con dos computadoras para el desarrollo y un servidor
 - Cliente:
 - Intel Core 2 Duo 3.00Ghz, 3GB de RAM, 250GB en HD

- Pantalla Plana de 17”
- Tarjeta de Red 10/100/1000
- Full Multimedia
- Servidor:
 - HP Proliant ML350 G6
 - Procesador: Intel® Xeon® E5506 (4 núcleos, 2,13 GHz, 4 MB L3, 80W)
 - 3 Discos de 146 GB en RAID 5
 - 2 Fuentes de Poder 750W
 - 2 Tarjetas de Red 10/100/1000
 - 12 GB de memoria RAM
- Software: Moodle en su versión 1.9 (ultimo release estable hasta el momento de desarrollo del proyecto), complementos descargables de la red desde la Web de Moodle.
- Información: Moodle cuenta con una amplia gama de información para sustentar el desarrollo de proyectos basados en su plataforma (www.moodle.org), se tienen también fuentes bibliográficas encontrables fácilmente en la Internet, así como la bibliografía anexada al final de este proyecto.

Estructura tecnológica del proyecto:

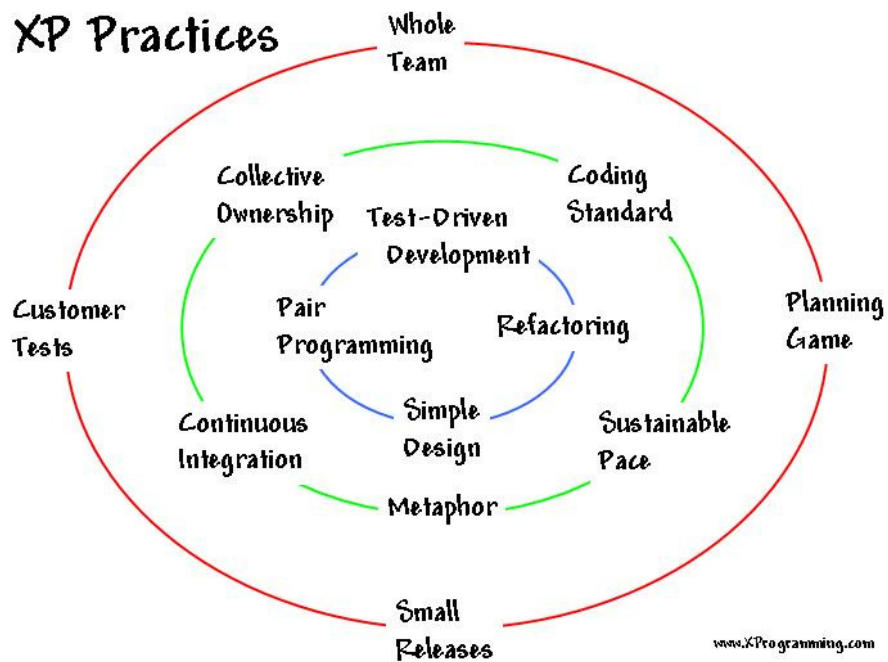
- Equipos:
 - Servidor: HP Proliant ML350 G6
 - Estaciones de desarrollo: Laptops para poder trabajar en cualquier sitio a cualquier hora a través de una conexión a Internet trabajando bajo Sistema Operativo Windows.
- Lenguaje de desarrollo: PHP
- Se ocuparán los servidores Web, y las Bases de datos de la escuela.
- El sistema debe estar disponible las 24 horas del día, los 7 días de la semana.

- Debe soportar un gran número de usuarios.

3.1.3.- DESCRIPCIÓN DE LA METODOLOGÍA DE DESARROLLO: EXTREME PROGRAMMING (XP)

El eXtreme Programming o Programación Extrema es una metodología ágil de desarrollo de software, que se basa en la simplicidad, el valor, la comunicación y la realimentación o reutilización del código desarrollado.

Gráfico 3.1. Diagrama de eXtreme Programming



Fuente: www.xprogramming.com

- **La simplicidad** ayuda a que los desarrolladores de software encuentren soluciones más simples a problemas, según el cliente lo estipula. Los desarrolladores también crean características en el diseño que pudieran ayudar a resolver problemas en un futuro.

- **La comunicación** prevalece en todas las prácticas de extreme programming. Comunicación cara a cara es la mejor forma de comunicación, entre los desarrolladores y el cliente. Método muy ágil. Gracias a esto el equipo esta pude realizar cambios que al cliente no le gustaron. También apoya agilidad con la extensión del conocimiento tácito dentro del equipo del desarrollo, evitando la necesidad de mantener la documentación escrita.
- **La retroalimentación** continua del cliente permite a los desarrolladores llevar y dirigir el proyecto en una dirección correcta hacia donde el cliente quiera.
- **El valor** requiere que los desarrolladores vayan a la par con el cambio, por que sabemos que este cambio es inevitable, pero el estar preparado con una metodología ayuda a ese cambio.

La metodología se diseña para entregar el software según las necesidades de cliente y cuando sea necesaria.

Extreme Programming promueve el trabajo del equipo. Cada integrante del proyecto (cliente, desarrolladores, etc.) forman parte integral del equipo encargado de desarrollar software de calidad.

3.1.3.1.-Prácticas de eXtreme Programming

Según los valores promovidos por XP se fundamenta en las siguientes trece prácticas:

1. The planning game
2. Test-driven development
3. Pair programming

4. Merciless refactoring
5. Simple Design
6. Colective code ownership
7. Continuos Integration
8. On-site Costumer
9. Small releases
- 10.40 hours Hjek
- 11.Apply coding standards
- 12.System metaphore
- 13.Stand-up meeting

A continuación definiremos cada uno de ellos:

- **The planning game (Juego de la Planificación).** Son una serie de actividades planificadas que son llamados juegos planificados por extreme programming, son asumidos durante el transcurso del proyecto.
- **Test-driven development (Pruebas).** Esta es la forma de prueba de unidad y de prueba de aceptación, donde las pruebas se escriben antes de la implementación del código. Las pruebas de aceptación, son escritas por el cliente, se automatizan y prueban generalmente la funcionalidad del sistema. Las pruebas de unidad son escritas por los desarrolladores y prueba que las unidades del código estén a un nivel según lo esperado.
- **Pair programming (Programación en Parejas).** Extreme programming requiere que todo el código de la producción sea producido por un par de desarrolladores compartiendo un monitor y un teclado en una estación de trabajo. Uno actúa como conductor y escribe código mientras que el otro, el navegador, observa qué está pasando y proporciona consejos estratégicos para resolver los problemas de manera eficiente. El conductor tiene que estar explicando que es lo que esta haciendo y se pueden intercambiar

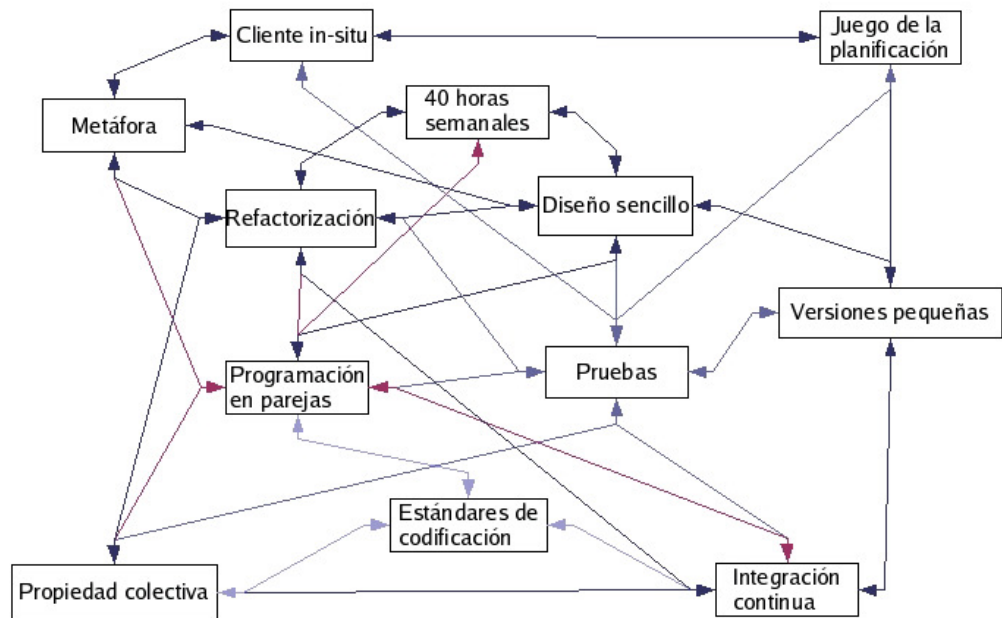
los roles con frecuencia. Por otro lado, para facilitar el trabajo entre los miembros del equipo, estos pueden interactuar con otros individuos para realizar la tarea.

- **Merciless refactoring (Refactorización).** Los desarrolladores requieren mejorar el código sin que este cambie su función para realizar un código simple. Esto dará oportunidad a modificarlo cuando haya necesidad de cambiar una característica.
- **Simple Design (Diseño Sencillo).** Los desarrolladores necesitan seguir el principio de YAGNI (You Are not Going to Need It) al producir un diseño para una historia dada. El diseño no debe incluir otra cosa que no sea necesaria para ejecutar la historia y pasar las pruebas de unidad.
- **Colective code ownership (Propiedad Colectiva).** Todos los desarrolladores son propietarios del código. Cualquier desarrollador puede modificar el código base en cualquier momento aunque haya sido hecho por otro par de desarrolladores. El objetivo es quitar dependencias en individuos.
- **Continuos Integration (Integración Continua).** Al cabo de un día el sistema deberá de ser integrado por una máquina, cada vez que un par de programadores tengan una clase ya probada unitariamente. Si al añadir la clase el sistema completo sigue funcionando correctamente, la tarea fue realizada con éxito. De lo contrario, esto permite que los desarrolladores detecten errores en la integración del sistema en una etapa antes ejecutando las pruebas de unidad del proyecto.
- **On-site Customer (Cliente In-Situ).** Los desarrolladores tienen acceso continuo con el cliente para poner en claro las historias y discutir el desarrollo de ediciones y proporcionar retroalimentación. Y un cliente real debe permanecer junto al equipo de desarrollo, para cualquiera duda o aclaración.
- **Small releases (Versiones Pequeñas).** Al final de cada iteración el sistema es lanzado y el cliente lo observa. Se lanza primero

unos meses antes de estar completamente terminado, las otras versiones serán mas frecuentes entre un día y un mes. La mayoría de estos lanzamientos es para conseguir retroalimentación del cliente.

- **40 Hours Week (40 Horas Semanales).** Si queremos que sea un trabajo de calidad , los trabajadores tienen que estar bien descansados , no superando la regla de las 40 horas semanales y que tengan otras cosas que hacer y no estén obsesionados con el trabajo.
- **Apply coding standards (Estándares de Codificación).** El código es la forma principal de documentación y por lo tanto debe de estar escrito de una manera clara y constante, que pueda ser identificado fácilmente por cualquier programador de otro equipo.
- **System metaphore (Metáfora).** Un lenguaje de metáforas se utiliza para describir la arquitectura del sistema. Esto ayuda a la comunicación entre los mismos desarrolladores y entre los desarrolladores y el cliente.
- **Stand-up meeting (Reunión).** Una reunión de 15 minutos en el comienzo de cada día para discutir problemas encontrados el día anterior y los problemas que se resolverán durante el día.

Gráfico 3.2. Prácticas de eXtreme Programming



Fuente: www.info-ab.uclm.es

3.1.3.2.-Roles de Extreme Programming

Programador.

El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

Cliente.

El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.

Encargado de pruebas (Tester)

El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

Encargado de seguimiento (Tracker)

El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

Entrenador (Coach)

Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

Consultor

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

Gestor (Big boss)

Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

3.1.3.3.-Proceso de eXtreme Programming

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo de Extreme Programming consiste en los siguientes pasos, presentados a grandes rasgos:

- a. El cliente define el valor de negocio a implementar.
- b. El programador estima el esfuerzo necesario para su implementación.
- c. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- d. El programador construye ese valor de negocio.
- e. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la

Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

3.1.3.3.1.- Fase I: Exploración

En esta fase, los clientes plantean de manera general las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

3.1.3.3.2.- Fase II: Planificación de la Entrega

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

3.1.3.3.3.- Fase III: Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el

cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

3.1.3.3.4.- Fase IV: Producción

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

3.1.3.3.5.- Fase V: Mantenimiento

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

3.1.3.3.6.- Fase VI: Muerte del Proyecto

Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

3.1.4.- DESCRIPCION DE LA METODOLOGIA DE DESARROLLO: SCRUM

Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

Scrum es una metodología ágil, y como tal:

- Es un modo de desarrollo de carácter adaptable más que predictivo.
- Orientado a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.

Se comienza con la visión general del producto, especificando y dando detalle a las funcionalidades o partes que tienen mayor prioridad de desarrollo y que pueden llevarse a cabo en un periodo de tiempo breve (normalmente de 30 días).

Cada uno de estos periodos de desarrollo es una iteración que finaliza con la producción de un incremento operativo del producto.

Estas iteraciones son la base del desarrollo ágil, y Scrum gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente.

3.1.4.1.- Características De Un Proceso Scrum

La metodología Scrum asume que el proceso de desarrollo de software es impredecible, y lo trata como a una “caja negra” controlada, en vez de manejarlo como un proceso completamente definido. Ésta es una de las

principales diferencias entre Scrum y otras metodologías, como los modelos de espiral o de cascada, en los cuales el proceso de desarrollo se define por completo desde el inicio. Por tratar de planificar el proceso en forma completa desde el principio, las metodologías tradicionales fallan al toparse con algunos problemas habituales del desarrollo de software, como la falta de comprensión de los requerimientos al empezar el proceso, el cambio en los requerimientos durante el proceso, o la dificultad para prever los resultados del uso de nuevas herramientas y tecnologías.

Otra diferencia de Scrum con las metodologías tradicionales es que no trata el proceso de desarrollo de software como un proceso lineal, en el que se sigue la secuencia de análisis, diseño, codificación y testing. En Scrum, el proyecto puede iniciarse con cualquier actividad, y cambiar de una a otra en cualquier momento.

Un proyecto administrado mediante Scrum se organiza en iteraciones, llamadas Sprints, que normalmente tienen entre dos y cuatro semanas de duración. Al principio de cada Sprint se establece una lista de requerimientos llamada backlog, que debe completarse cuando éste finalice. A diario se realizan breves reuniones del equipo de desarrollo, en las que se exponen los avances y los problemas encontrados, y se señalan posibles caminos para resolverlos (la resolución detallada de estos problemas no debe determinarse durante la reunión, para mantener su brevedad).

3.1.4.2.-Proceso, Roles Y Reuniones De Scrum

3.1.4.2.1.- *Los Roles*

En Scrum, el equipo se centra en construir software de calidad. La gestión de un proyecto Scrum tiene su objetivo en definir cuáles son las

características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en vencer cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo.

El equipo Scrum está formado por los siguientes roles:

- **Scrum master:** Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el Product Owner para maximizar el ROI¹⁴ (Return Over Investment).
- **Product owner (PO):** Representante de los accionistas y clientes que usan el software. Se focaliza en la parte de negocio y el es responsable del retorno de la inversión del proyecto (entregar un valor superior al dinero invertido). Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en el Product Backlog y las reprioriza de forma regular.
- **Team:** Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las historias a las que se comprometen al inicio de cada Sprint.

Cabe recalcar que se puede nombrar al rol de “Otros interesados” como por ejemplo la Dirección General, la Dirección Comercial, Marketing, Usuarios, etc. Pero los roles expuestos anteriormente son los utilizados en Scrum aplicado al desarrollo de Software.

3.1.4.2.2.- El proceso

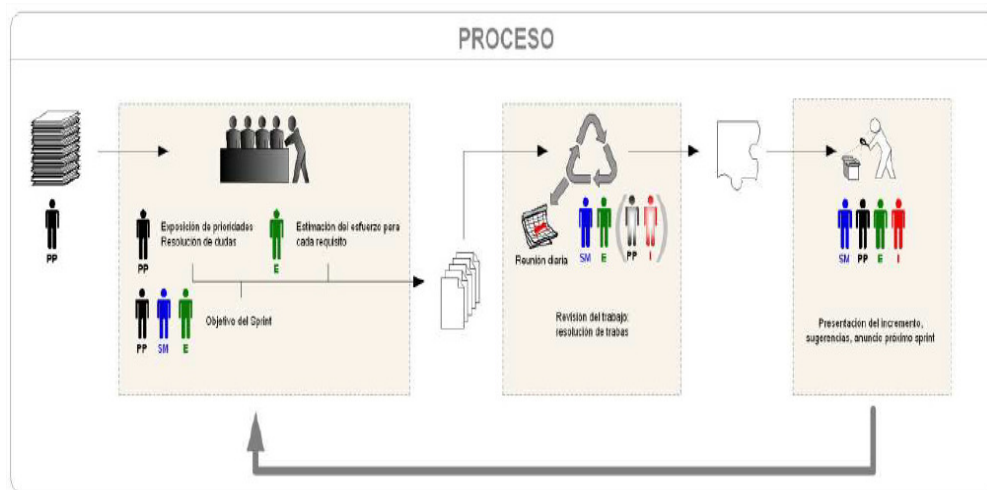
El desarrollo se realiza de forma iterativa e incremental. Cada iteración nos arroja como resultado una versión del software con nuevas

¹⁴ ROI.- Porcentaje que se calcula en función de la inversión y los beneficios obtenidos, para obtener el ratio de retorno de inversión.

prestaciones listas para ser usadas. En cada nuevo Sprint (iteración), se va ajustando la funcionalidad ya construida y se añaden nuevas prestaciones priorizándose siempre aquellas que aporten mayor valor de negocio.

- **Product Backlog:** Conjunto de requisitos denominados historias descritos en un lenguaje no técnico y priorizados por valor de negocio, o lo que es lo mismo, por retorno de inversión considerando su beneficio y coste. Los requisitos y prioridades se revisan y ajustan durante el curso del proyecto a intervalos regulares.
- **Sprint Planning:** Reunión durante la cual el Product Owner presenta las historias del backlog por orden de prioridad. El equipo determina la cantidad de historias que puede comprometerse a completar en ese Sprint, para en una segunda parte de la reunión, decidir y organizar cómo lo va a conseguir.
- **Sprint:** Iteración de duración prefijada durante la cual el equipo trabaja para convertir las historias del Product Backlog a las que se ha comprometido, en una nueva versión del software totalmente operativo.
- **Sprint Backlog:** Lista de las tareas necesarias para llevar a cabo las historias del Sprint.
- **Daily Sprint meeting:** Reunión diaria de cómo máximo 15 min. en la que el equipo se sincroniza para trabajar de forma coordinada. Cada miembro comenta que hizo el día anterior, que hará hoy y si hay impedimentos.
- **Demo y retrospectiva:** Reunión que se celebra al final del Sprint y en la que el equipo presenta las historias conseguidas mediante una demostración del producto. Posteriormente, en la retrospectiva, el equipo analiza qué se hizo bien, qué procesos serían mejorables y discute acerca de cómo perfeccionarlos.

Gráfico 3.3. Proceso de SCRUM



Fuente: www.navegapolis.net

3.1.4.2.3.- Las Reuniones

Intervienen en el proceso de Scrum con el objetivo de supervisar los avances del desarrollo del proyecto al ser iterativa e incremental la metodología aplicada, discutir los distintos cambios que se deban realizar (cambios de requisitos, etc.), todo esto dividiéndose en una serie de reuniones como las siguientes:

- **Daily Scrum.** Cada día de un Sprint, se realiza la reunión sobre el estado de un proyecto. Esto se llama "daily stand up". La reunión comienza puntualmente a su hora. A menudo hay castigos acordados por el equipo para quien llegue tarde. La reunión tiene una duración fija de 15 minutos, de forma independiente del tamaño del equipo. Todos los asistentes deben mantenerse de pie (esto ayuda a mantener la reunión corta). La reunión debe ocurrir en la misma ubicación y a la misma hora todos los días.
- **Scrum de Scrum.** Se efectúa normalmente cada día después del "Daily Scrum". Este tipo de reuniones permiten a los grupos de

equipos discutir su trabajo, enfocándose especialmente en áreas de solapamiento e integración. Asiste una persona asignada por cada equipo. Se maneja de manera similar al Daily Scrum además de agregarle preguntas puntuales sobre el desarrollo de cada equipo.

- **Sprint Planning Meeting.** Se lleva a cabo una “Reunión de Planificación del Sprint”, al inicio del ciclo Sprint (cada 15 o 30 días). Se realiza para seleccionar qué trabajo se hará, para preparar, con el equipo completo, el Sprint Backlog que detalla el tiempo que tomará hacer el trabajo, también se hace para identificar y comunicar cuánto del trabajo es probable que se realice durante el actual Sprint. Tiene como límite 8 horas de duración.

Al final del ciclo Sprint, dos reuniones se llevaran a cabo: la “Reunión de Revisión del Sprint” y la “Retrospectiva del Sprint”.

- **Reunión de Revisión del Sprint.** Tiene una duración límite de 4 horas. Se la realiza para: revisar el trabajo que fue completado y no completado, presentar el trabajo completado a los interesados (demo). Hay que recordar que el trabajo incompleto no puede ser demostrado.
- **Retrospectiva del Sprint.** Después de cada Sprint, se lleva a cabo una retrospectiva del Sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el Sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso. Esta reunión tiene un tiempo fijo de cuatro horas.

Gráfico 3.4. Algunas Reuniones de SCRUM



Fuente: www.navegapolis.net

3.1.5.- COMPARATIVA DE LAS METODOLOGÍAS SCRUM Y XP

En relación a las mejores prácticas de cada una de las metodologías descritas anteriormente, se puede concluir y garantizar la calidad del software. Luego del respectivo análisis individual de las metodologías propuestas que por lo citado anteriormente son las más adaptables al desarrollo de nuestro proyecto, nos enfocaremos en esta sección en realizar una pequeña comparativa de dichas metodologías.

3.1.5.1.- Semejanzas

Resaltaremos las semejanzas más imponentes.

- Ambas son metodologías de desarrollo ágiles basadas en los valores del “Agile Manifesto”
- Ambas utilizan historias de usuario.

- Realizan continuamente entregas al cliente en cortos periodos de tiempo.
- Los tipos de reuniones en ambas metodologías son de tipo rápido y concreto.

3.1.5.2.-Diferencias

En la siguiente tabla se podrán apreciar algunas de las diferencias de las metodologías.

Tabla 3.2. Diferencias SCRUM y Xp¹⁵

SCRUM	EXTREME PROGRAMMING
Las iteraciones de entrega son de dos a cuatro semanas y se conocen como Sprint	Las iteraciones de entrega son de una a tres semanas
Al finalizar un Sprint, las tareas que se han realizado del Sprint Backlog y en las que el Product Owner ha demostrado su conformidad ya no se vuelven a tocar en ningún momento.	Las tareas que se van terminando en las diferentes entregas son susceptibles a modificaciones durante el transcurso de todo el proyecto, incluso después de que funcionen correctamente
Cada miembro del Scrum Team trabaja de manera individual	Los miembros trabajan en parejas
El Scrum Team trata de seguir el orden de prioridad que marca el Product Owner en el Sprint Backlog pero si ven que es mejor modificar el orden de prioridad para el	El equipo de desarrollo sigue estrictamente el orden de prioridad de las tareas definido por el cliente (aunque el equipo de desarrollo les ayude a decidir,

¹⁵ Maestría en Ingeniería de Sistemas con Mención en Tecnología de la Información. Lima-2007

desarrollo de las tareas, pueden hacerlo	ellos son los que mandan)
El Scrum es una metodología de desarrollo ágil más basada en la administración del proyecto.	XP se centra más en la propia programación o creación del producto

Elaborado por: PROAÑO, Marco

Al conocer las semejanzas y diferencias entre las metodologías propuestas y basándonos también en los estudios realizados por Henrik Kniberg en su libro “Scrum y Xp desde las Trincheras” en el año 2016 y en los estudios de Ken Schwaber y Mike Beedle “Agile Software Development with Scrum” en el año 2017, nos encontramos con la particularidad de que ambas metodologías pueden ser adaptables y complementarias entre sí, por tal motivo analizaremos la posibilidad de trabajar bajo la conjunción de ambas metodologías.

3.1.6.- SCRUM Y XP

Scrum y Extreme Programming proveen prácticas y reglas complementarias entre sí. Se superponen por ejemplo en el Planning Game (XP) y el Sprint Planning (SCRUM). Ambos poseen valores similares, reduciendo al mínimo los problemas de conexión entre la dirección y los desarrolladores. Combinados proveen una estructura dentro de la cual el cliente pueda desarrollar un producto que mejor se adapte a sus necesidades, y puede poner en práctica la funcionalidad, compatibilidad, incrementabilidad para tomar mejor ventaja de las oportunidades de negocio. A continuación se listan las prácticas que facilitan esta funcionalidad:

16 <http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf>
17 <http://faculty.salisbury.edu/~xswang/Research/Papers/SERelated/scrum/ScrumXP.pdf>

- **Iteraciones.** Todo el trabajo es hecho iterativamente, con el cliente siendo capaz de dirigir el proyecto y cada iteración del mismo.
- **Incrementos.** Cada iteración produce un incremento de la funcionalidad de más alta prioritaria del cliente. De ser deseado, el cliente puede dirigir a los desarrolladores para convertirse en estos incrementos en incrementos de funcionalidad viva y operacional en cualquier momento.
- **Emergencia.** Sólo aquella funcionalidad que el cliente ha seleccionado para la siguiente iteración es considerada y construida. El cliente no paga por la funcionalidad que él o ella no podrían seleccionar, y los reveladores no tienen que codificar, eliminar fallos, y mantener el código irrelevante.
- **Auto-Organización.** El cliente dice lo que él o ella quieren; el desarrollo determina cuánto se puede desarrollar durante una iteración y descifran las tareas a realizar.
- **Colaboración.** El negocio y la ingeniería colaboran sobre como mejor construir el producto y lo que el producto debería hacer entre iteraciones.

3.1.7.- PROCESO SCRUM CON PRÁCTICAS DE INGENIERÍA XP

La frase “El Proceso Scrum con Prácticas de Ingeniería XP“, se refiere al empleo de Scrum para manejar los pasos tomados para desarrollar el software, en la conjunción con el empleo de XP para asegurar la calidad del software.

Estos son algunos procesos Scrum que se pueden utilizar dentro de un proyecto de desarrollo de Software:

- Sprints y Sprint planning meetings.
- Daily Scrums
- Product backlog and product owner

- Sprint backlog
- ScrumMaster

Estos son algunas de las prácticas XP que se pueden utilizar dentro de un proyecto de desarrollo de Software:

- Diseño Simple
- Pruebas
- Integración Continua
- Refactorización
- Programación por Parejas

El empleo del proceso Scrum con prácticas de Ingeniería XP tiene un impacto significativo sobre la productividad del equipo de proyecto. La adopción de cualquiera de estas metodologías tendrá un impacto beneficioso sobre un equipo de proyecto. Sin embargo el uso conjunto de estas metodologías inculca prácticas y valores en la dirección, clientes, y el desarrollo que les permite crear un equipo unificado y disciplinado. Por tanto se concluye que se utilizarán estas dos metodologías como complementarias; teniendo un impacto significativo tanto sobre la productividad del equipo como sobre la calidad del proyecto de desarrollo que será entregado.

3.1.8.- APLICACIÓN DE SCRUM Y EXTREME PROGRAMMING EN EL PROYECTO

Según el análisis efectuado en las secciones anteriores, se optará por la programación extrema XP conjuntamente trabajando con Scrum, ya que éstas se ajustarán mejor a las características del proyecto.

Más concretamente, se cumplen las características semejantes en el empleo de XP con Scrum en un proyecto:

- Interés sincero por todas las partes en que el proyecto tenga éxito.
- El equipo de trabajo es pequeño.
- No existe un contrato fijo previo especificando tiempo, recursos y alcance.
- El equipo dispone de una formación elevada (esa es la finalidad) y capacidad de aprender.
- El proyecto tiene un riesgo alto en cuanto a lo innovador de la tecnología.
- El proyecto sufrirá cambios de requerimientos a o largo de su desarrollo.

También se ha tenido en cuenta el éxito que ha tenido el empleo de XP con Scrum en proyectos Open Source como es el caso de Moodle, cuyas características hacen que se adapte especialmente bien estas metodologías.

Como es evidente no todas las prácticas y procesos son aplicables al presente trabajo, ya que se trata de un proyecto de titulación; por lo que a continuación se especifica aquellos aplicables en este proyecto:

3.1.8.1.-Sprints y Sprint Planning Meetings.

Se ha decidido realizar cuatro releases con sus respectivas iteraciones. Cada una de ellas proporcionando un valor de negocio claro, a grosso modo serán:

- Departamentos: gestión de departamentos o programas académicos
- Períodos Académicos: gestión de períodos académicos
- Calificaciones: gestión de calificaciones
- Módulos: gestión de módulos

- Tareas: gestión de tareas entregables y recibidas.

En el Capítulo 3, se detallará la Planificación de las entregas en cada una de estas iteraciones.

3.1.8.2.-Product Backlog y Product Owner

Entre un representante de las TIC's y el graduando, se realizará el Product Backlog siendo para este proyecto el departamento en si de las TIC's de la ESPE-L el Product Owner.

3.1.8.3.-Sprint Backlog

Se realizará dentro de cada Sprint planning meeting para mantener el registro de las actividades a realizarse.

3.1.8.4.-Scrum Master

Para nuestro proyecto el director y codirector de la tesis se convertirán en Scrum Masters.

3.1.8.5.-Diseño Simple

El diseño se ha mantenido sencillo, desde luego pasando los tests, sin código duplicado y con un mínimo de código gracias al núcleo del sistema de Moodle que proporciona un gran dinamismo y evita la necesidad de implementación de operaciones repetitivas.

3.1.8.6.-Pruebas - Tests

Una de las principales aportaciones de esta metodología es el concepto de desarrollo dirigido por tests (Test Driven Development). Los tests son

realizados a priori con el fin de prevenir errores, no de solucionarlos. Esto confiere una gran calidad al software resultante. Los test serán realizados durante el desarrollo del proyecto a través de las pruebas de unidad.

3.1.8.7.-Integración Continua

Las pruebas de integración se efectuarán siempre, antes de añadir cualquier nueva clase al proyecto, o después de modificar cualquiera existente.

3.1.8.8.-Refactorización

En las sucesivas iteraciones será necesario refactorizar partes del núcleo del sistema Moodle, a lo que contribuirá en gran parte la cantidad de tests realizados.

3.1.8.9.-Programación por Parejas

Se aplicará conjuntamente con un delegado del área de las TIC's de la escuela, tomándose en cuenta que se referirá como "Desarrollo por Parejas", al ser Moodle una plataforma con un mínimo o nulo porcentaje de desarrollo por programación.

3.1.8.10.- Releases Pequeños

Se seguirá esta práctica, liberando nuevas versiones según la funcionalidad que se vaya implementando.

CAPÍTULO 4.

4. DESARROLLO DEL SISTEMA

Una vez que se han investigado los lineamientos de la Programación Extrema y SCRUM, en capítulos anteriores se desarrollará y/o implementará nuevas funcionalidades o módulos en un LMS Open Source Moodle.

De acuerdo con la metodología, XP tiene las siguientes fases: planeación o análisis, diseño, implementación ó codificación, y pruebas, pero estas fases se aplican en cada iteración de SCRUM, es decir al resolver cada una de las historias de usuario, se hace el diseño, se codifica y se prueba hasta que el sistema soporte cada uno de los requerimientos del cliente. Es por este motivo que en este capítulo no se podrá distinguir cada una de las fases como un título por separado ya que la forma de trabajar de XP y SCRUM sugieren realizar los pasos antes mencionados.

En primera instancia definiremos el problema como tal, estableciendo de esta manera los requerimientos del cliente. Estos requerimientos fueron tomados de una reunión informal. A continuación se detallan cada una de las fases de desarrollo hasta obtener la plataforma funcional.

4.1.- DEFINICION DEL PROBLEMA

En la sección 1.3 y 1.4 se realizó un análisis sobre el LMS que se seleccionó. En la sección 1.3 el análisis se basó en la arquitectura y metodología de desarrollo de Moodle en su versión 1.9, y por otro lado en la sección 1.4, se identificaron las diferentes características con las que

cuenta actualmente la plataforma. Es preciso mencionar que desde la versión 1.5.x de Moodle se tienen versiones totalmente estables.

Además, tomando en cuenta que este proyecto tiene como objetivo apoyar el proceso de enseñanza-aprendizaje en programas de pregrado y postgrado de la Escuela Superior Politécnica del Ejército Sede Latacunga. En este sentido, se ve necesario que dentro la plataforma existan módulos de gestión de carreras o programas académicos de pregrado y postgrado.

Cabe recalcar, que como herramienta o componente de un curso es necesario la incorporación de un módulo de publicación de calificaciones en diferentes categorías (pruebas, exámenes, exposiciones, etc.) por parte del profesor.

Es preciso mencionar que con la implementación de estos nuevos módulos, este proyecto no pretende ser un Sistema Administrativo para la Institución de, sino ofrecer al usuario un conjunto de “herramientas en línea”, que apoyen el proceso de enseñanza-aprendizaje en programas de pregrado y postgrado de la Escuela Superior Politécnica del Ejército Sede Latacunga.

4.2.- ANALISIS (PREPARACIÓN DEL PROYECTO)

Para proceder con el desarrollo de este proyecto, se mantuvo reuniones con el cliente y colaboradores, que en nuestro caso fueron los responsables del departamento de las TIC's de la Escuela Politécnica del Ejército Sede Latacunga, con quienes a través de conversaciones se pudieron establecer los requerimientos o necesidades que necesitaba la plataforma virtual de la institución.

4.2.1.- ROLES DEL PROCESO DE DESARROLLO

En esta sección asignaremos los roles de desarrollo dentro del proceso, estos roles se han definido dentro de la sección 2.1.4.2 de este documento, así tenemos:

- **Scrum Master:** Ing. Fabián Montaluisa (Director del Proyecto), Ing. Javier Montaluisa (Codirector)
- **Product Owner:** Departamento TIC's ESPE-L
- **Developers Team:** Marco Proaño, Mónica Llumitasig

4.2.2.- HISTORIAS DE USUARIO (XP)

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software, estas equivalen al Product Backlog en SCRUM, en las cuales se describen brevemente las características que el sistema debe tener desde la perspectiva del cliente.

Para tener una base más centrada y concreta del esfuerzo necesario se aplicarán las historias de usuario de XP, las estimaciones se establecerán utilizando como medida el punto. Un punto, equivale a una semana ideal de desarrollo, y las historias generalmente valen de 1 a 3 puntos.

Para este proyecto, las historias de usuarios son definidas en base a la definición del problema expuesto en el subcapítulo 3.1.

4.2.2.1.-Historia 1: Creación de Carreras o Programas Académicos

Tabla 4.1. Historia 1 - Creación de carreras o programas académicos

HISTORIA DE USUARIO	
Numero: 1	Usuario: Administrador de la plataforma
Nombre de historia: Creación de carreras o programas académicos	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Baja (Alta / Media / Baja)
Puntos estimados: 2	Iteración asignada: 1
Desarrollador responsable: Mónica Llumitasig, Marco Proaño	
Descripción: Para la creación de una nueva carrera o programa académico el Administrador de la Plataforma, debe ingresar los datos que se pidan en el formulario y agregar la descripción breve de la carrera incluyendo datos como la modalidad de estudios (presencial, semi-presencial o distancia), la duración, etc.	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.2.2.2.-Historia 2: Visualización, Modificación, Eliminación y Búsqueda de Carreras o Programas Académicos

Tabla 4.2. Historia 2 - Visualización, modificación, eliminación y búsqueda de carreras o programas académicos

HISTORIA DE USUARIO	
Numero: 2	Usuario: Administrador de la plataforma
Nombre de historia: Visualización, modificación, eliminación y búsqueda de carreras o programas académicos	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Baja (Alta / Media / Baja)
Puntos estimados: 1	Iteración asignada: 1
Desarrollador responsable: Mónica Llumitasig, Marco Proaño	
Descripción: Al usuario se le presenta un listado de carreras o programas académicos con información la información que esta haya sido ingresada. El usuario administrador podrá visualizar, modificar, eliminar las carreras registradas.	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.2.2.3.-Historia 3: Gestión de Cursos Académicos

Tabla 4.3. Historia 3 - Gestión de Cursos Académicos

HISTORIA DE USUARIO	
Numero: 3	Usuario: Administrador de la plataforma
Nombre de historia: Gestión de Cursos Académicos	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Baja (Alta / Media / Baja)
Puntos estimados: 3	Iteración asignada: 2
Desarrollador responsable: Mónica Llumitasig, Marco Proaño	
Descripción: El sistema debe permitir el registro y visualización de la información (planificación) de los cursos académicos de una carrera o programa académico. Esta información puede ser susceptible de modificación o eliminación, así como estar disponible para su búsqueda.	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.2.2.4.-Historia 4: Gestión de Recursos del Curso

Tabla 4.4. Historia 4 - Gestión de Recursos del Curso

HISTORIA DE USUARIO	
Numero: 4	Usuario: Administrador de la plataforma, Profesores
Nombre de historia: Gestión de Recursos del Curso	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Baja (Alta / Media / Baja)
Puntos estimados: 4	Iteración asignada: 2
Desarrollador responsable: Mónica Llumitasig, Marco Proaño	
Descripción: El sistema debe permitir el registro y visualización de la información y materiales de un curso creado con anterioridad. Esta información puede ser susceptible de modificación o eliminación.	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.2.2.5.-Historia 5: Gestión de Registro de Estudiante

Tabla 4.5. Historia 5 - Gestión de Registro de Estudiante

HISTORIA DE USUARIO	
Numero: 5	Usuario: Administrador de la plataforma, Profesores, Estudiantes
Nombre de historia: Gestión de Registro de Estudiante	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Baja (Alta / Media / Baja)
Puntos estimados: 2	Iteración asignada: 3
Desarrollador responsable: Mónica Llumitasig, Marco Proaño	
Descripción: El sistema debe permitir el registro y confirmación del registro de un estudiante en un curso, mediante el envío de confirmación y aprobación de registro por parte del profesor (en algunos casos con una clave proporcionada con anterioridad). Esta información puede ser susceptible de modificación o eliminación.	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.2.2.6.-Historia 6: Gestión de Calificaciones

Tabla 4.6. Historia 6 - Gestión de Calificaciones

HISTORIA DE USUARIO	
Numero: 6	Usuario: Profesores
Nombre de historia: Gestión de Calificaciones	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Baja (Alta / Media / Baja)
Puntos estimados: 2	Iteración asignada: 3
Desarrollador responsable: Mónica Llumitasig, Marco Proaño	
Descripción: El sistema debe permitir el registro y consulta de las calificaciones parciales de los estudiantes que pertenecen a un curso dentro de un período académico determinado mediante la revisión directa dentro del curso. También debe permitir el registro y consulta de las calificaciones finales (calificación de aprobación del curso del estudiante) y el tipo de aprobación (exonerado, aprobado, fallido, etc.).	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.2.2.7.-Historia 7: Gestión de Perfil de Usuario

Tabla 4.7. Historia 7 – Gestión de perfil de usuario

HISTORIA DE USUARIO	
Numero: 7	Usuario: Estudiantes, Profesores
Nombre de historia: Gestión de Perfil de Usuario	
Prioridad en negocio: Media (Alta / Media / Baja)	Riesgo en desarrollo: Baja (Alta / Media / Baja)
Puntos estimados: 1	Iteración asignada: 4
Desarrollador responsable: Mónica Llumitasig, Marco Proaño	
Descripción: El sistema debe permitir el registro y administración del perfil personal de cada usuario del sistema (Datos personales, intereses, blog personal, etc.), así también le permitirá al usuario manejar sus mensajes privados ya sean de compañeros o administradores.	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.2.2.8.-Historia 8: Diseño estético de la Plataforma

Tabla 4.8. Historia 8 – Gestión de Récord Académico por Curso

HISTORIA DE USUARIO	
Numero: 8	Usuario: Desarrolladores
Nombre de historia: Diseño estético de la plataforma	
Prioridad en negocio: Baja (Alta / Media / Baja)	Riesgo en desarrollo: Baja (Alta / Media / Baja)
Puntos estimados: 1	Iteración asignada: 4
Desarrollador responsable: Mónica Llumitasig, Marco Proaño	
Descripción: Los desarrolladores del sistema, de acuerdo con las especificaciones de combinaciones en colores y en forma de diseño estético de la plataforma Web de la escuela (http://webltga.espe.edu.ec), elegirán la plantilla mas adecuada para la plataforma virtual realizando los cambios necesarios.	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.2.3.- PRODUCT BACKLOG (SCRUM)

Como lo habíamos mencionado anteriormente el Product Backlog¹⁸ es la lista primaria de los requerimientos del usuario con estimaciones de tiempo y asignaciones, los cuales obtenemos desde la lista de Historias de Usuario, así tenemos:

Gráfico 4.1. Gráfica de Grantt Plataforma Virtual ESPE-L

Project / Task Group / Task Name	Status	Assigned To	Start Date	Due Date	Completed Status	Modified By	Author	Archive
Plataforma Virtual ESPE-L	Active		10/01/2011	23/05/2011 0:00	0%	Admin	Admin	
Iteración 1	Overdue		10/01/2011	28/02/2011 23:51	0%	Admin	Admin	
Historia 1: Creación de Carreras o Programas Académicos	Overdue	Monica	10/01/2011 23:	24/01/2011 7:43	0%	Admin	Admin	
Historia 1: Creación de Carreras o Programas Académicos	Overdue	Marco	10/01/2011 23:	24/01/2011 7:43	0%	Admin	Admin	
Historia 2: Visualización, Modificación, Eliminación de Carrera	Overdue	Monica	07/02/2011 23:	28/02/2011 23:51	0%	Admin	Admin	
Historia 2: Visualización, Modificación, Eliminación de Carrera	Overdue	Marco	07/02/2011 23:	28/02/2011 23:51	0%	Admin	Admin	
Iteración 2	Overdue		07/02/2011	28/03/2011 8:19	0%	Admin	Admin	
Historia 3: Gestión de Cursos Académicos	Overdue		07/02/2011	28/02/2011 8:08	0%	Admin	Admin	
Búsqueda de un Curso Académico	Overdue	Monica	09/02/2011 0:1	14/02/2011 0:10	0%	Admin	Admin	
Eliminación de un Curso Académico	Overdue	Marco	22/02/2011 0:0	28/02/2011 8:08	0%	Admin	Admin	
Modificación de un Curso Académico	Overdue	Marco	15/02/2011 0:0	21/02/2011 8:06	0%	Admin	Admin	
Registro (Ingreso a la Plataforma) de Cursos Académicos	Overdue	Marco	07/02/2011 23:	14/02/2011 23:58	0%	Admin	Admin	
Historia 4: Gestión de Recursos del Curso	Overdue		01/03/2011	28/03/2011 8:19	0%	Admin	Admin	
Eliminación de Recursos del Curso	Overdue	Marco	15/03/2011 0:1	21/03/2011 8:17	0%	Admin	Admin	
Incorporación de Recursos al Curso	Overdue	Marco	01/03/2011 0:1	07/03/2011 0:13	0%	Admin	Admin	
Modificación de Recursos Académicos del Curso	Overdue	Monica	08/03/2011 0:1	14/03/2011 0:15	0%	Admin	Admin	
Visualización de Recursos del Curso	Overdue	Monica	22/03/2011 0:1	28/03/2011 8:19	0%	Admin	Admin	
Iteración 3	Overdue		04/04/2011	02/05/2011 0:24	0%	Admin	Admin	
Historia 5: Gestión de Registro de Estudiante	Overdue	Marco	04/04/2011 0:2	18/04/2011 0:21	0%	Admin	Admin	
Historia 6: Gestión de Calificaciones	Overdue	Monica	19/04/2011 0:2	02/05/2011 0:24	0%	Admin	Admin	
Iteración 4	Overdue		09/05/2011	23/05/2011 8:28	0%	Admin	Admin	
Historia 7: Gestión de Perfil de Usuario	Overdue	Marco	09/05/2011 0:2	16/05/2011 0:26	0%	Admin	Admin	
Historia 8: Diseño Estético de la Plataforma	Overdue	Monica	17/05/2011 0:2	23/05/2011 8:28	0%	Admin	Admin	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

18 http://www.dosideas.com/wiki/Backlog_Del_Producto

Como se puede apreciar, se ha realizado la estimación del tiempo de duración para cada iteración dentro del proceso de desarrollo del sistema con todas las tareas por completar (Estado Rojo - Pendiente).

Cabe recalcar que para la construcción del Product Backlog y la mayoría de elementos del proceso se utilizó el Software Team Task Manager¹⁹ en su versión 2.11.

4.3.- PLANIFICACIÓN DE ENTREGAS (SPRINT PLANNING)

Las historias de usuario servirán para crear el plan estimado de entregas. Para ello se convocó a una reunión con el cliente y los desarrolladores para crear el plan de entregas. El plan de entregas se usará para crear los planes de iteración.

Con cada historia de usuario descrita y evaluada en tiempo de desarrollo ideal, el cliente las agrupará en orden de importancia.

De esta manera, se procederá a trazar el plan de entregas en función de dos parámetros: Tiempo de desarrollo ideal (Esfuerzo) y Grado de importancia para el cliente (prioridad), para ello es necesario realizar una sinopsis de las historias de usuarios en base a la prioridad o esfuerzo.

¹⁹ <http://www.orbisoft.com/>

Tabla 4.9. Sinopsis de las Historias de usuarios

HISTORIA	PRIORIDAD	ESFUERZO
Creación de Carreras o Programas Académicos	Alta	2
Visualización, Modificación, Eliminación de Carreras o Programas Académicos	Alta	1
Gestión de Cursos Académicos	Alta	3
Gestión de Recursos del Curso	Alta	4
Gestión de Registro de Estudiante	Alta	2
Gestión de Calificaciones	Alta	2
Gestión de Perfil de Usuario	Media	1
Diseño Estético de la Plataforma	Media	1
Total de Semanas (Puntos)		16

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

En base a la tabla, se efectuó un resumen de las historias de usuario en relación a la prioridad y esfuerzo de cada una de ellas. En esta tabla se puede apreciar que el esfuerzo estimado es de 16 semanas.

Tabla 4.10. Sprint Planning

Iteraciones	Historias	Prioridad	Esfuerzo	Fecha Inicio	Fecha Final
1	Historia 1	Alta	2	10/01/2011	31/01/2011
	Historia 2	Alta	1		
2	Historia 3	Alta	3	07/02/2011	28/03/2011
	Historia 4	Alta	4		
3	Historia 5	Alta	2	04/04/2011	02/05/2011
	Historia 6	Alta	2		
4	Historia 7	Media	1	09/05/2011	23/05/2011
	Historia 8	Media	1		

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.3.1.- HERRAMIENTAS

En el desarrollo de este proyecto se han utilizado las siguientes herramientas en cada ámbito. Todo el software utilizado es gratuito solamente las herramientas hardware son costosas, pero pertenecientes a la escuela.

4.3.1.1.-Desarrollo

La plataforma, fue desarrollada bajo los siguientes aspectos técnicos:

- Equipos:
 - Servidor HP Proliant ML350 G6
 - Estaciones de desarrollo: Laptops trabajando bajo Sistema Operativo Windows.
- Lenguaje de desarrollo: PHP
- Servidores Web, Bases de datos.
- Navegadores web: Firefox, MS Internet Explorer, Google Chrome, Safari, Opera.
- phpMyAdmin en su ultima versión
- Apache en su ultima versión
- PHP en su ultima versión

4.4.- DESARROLLO ITERATIVO

Una vez definidas las historias de usuarios en la sección 3.2.1 y la planificación de entregas en la sección 3.2.2 se procederá a detallar cada una de las fases a seguirse en cada una de las iteraciones.

Se debe tomar en cuenta que no se realizan las reuniones de Scrum diarias por la premura del tiempo, además de las obligaciones laborales extra proyecto que los desarrolladores y director poseen, por tal motivo se

actualizan el Sprint Backlog, Burn Down Chart Y la lista de tareas pendientes y culminadas solamente al final de cada iteración en la revisión del progreso de estas.

4.4.1.- ITERACIÓN (SPRINT) 1

4.4.1.1.- Sprint Backlog²⁰

En esta fase se detallan las tareas y sub-tareas contenidas dentro de la primera iteración que engloba las historias de creación de carreras o programas académicos y de visualización, modificación, eliminación y búsqueda de carreras o programas académicos, las cuales tienen su tiempo y delegado de desarrollo, con lo cual se llevará constancia de las tareas que van desarrollándose con normalidad y en las que se tienen problemas.

Tabla 4.11. Sprint Backlog – Iteración 1

HIST.	TAREA	DELEG.	ESTADO	DIAS	1-7	8-14	15-21
				HRS PND.	504	336	168
1	Obtener lista de carreras	M&M	Term.		X		
1	Diseño presentación lista	M&M	Term.		X	X	
1	Dividir Categorías (Pregado/Posgrado)	M&M	Term.		X		
1	Creación de categorías	M&M	Term.		X		
1	Ingreso de carreras en categorías madre	M&M	En Progreso.		X	X	X
2	Diseño de módulo de búsqueda	M&M	Term.			X	
2	Establecer criterios de búsqueda	M&M	Term.			X	

²⁰ <http://www.proyectosagiles.org/lista-tareas-iteracion-sprint-backlog>

2	Formulario de carrera	M&M	Term.			X	X
2	Módulo de eliminación	M&M	Term.			X	X
2	Módulo de modificación	M&M	Term.			X	X

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.1.2.-Revisión del Sprint

Se presentan las listas de tareas realizadas y pendientes de cada desarrollador (generadas en Team Task Manager) del sistema para seguir midiendo su avance para llegar a la fecha límite planteada.

4.4.1.2.1.- Tareas completadas

Se enlistan las tareas que el desarrollador ha terminado hasta el momento de la revisión del Sprint en la reunión.

Desarrollador: Marco Proaño

Gráfico 4.2. Tareas Completadas Marco - Iteración 1

Task Assigned To:	Marco	Task Status:	Completed	Task Count:	2
Task Name	Project Name	Task Group Name	Modified Date	Author	
Historia 1: Creación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:00	Admin	
Historia 2: Visualización, Modificación, Eliminación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:02	Admin	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

Desarrollador: Mónica Llunitasig

Gráfico 4.3. Tareas Completadas Mónica - Iteración 1

Task Assigned To: Monica Task Status: Completed Task Count: 2

Task Name	Project Name	Task Group Name	Modified Date	Author
Historia 1: Creación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:01	Admin
Historia 2: Visualización, Modificación, Eliminación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:02	Admin

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.1.2.2.- Tareas pendientes

Se enlistan las tareas que el desarrollador tiene aun que realizar del total de tareas del Product Backlog hasta el final del proyecto.

Desarrollador: Marco Proaño

Gráfico 4.4. Tareas Pendientes Marco - Iteración 1

Task Assigned To:	Marco	Task Status:	Pending	Task Count:	7
Task Name	Project Name	Task Group Name	Modified Date	Author	
Eliminación de un Curso Académico	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	02/02/2011 22:05	Admin	
Modificación de un Curso Académico	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	02/02/2011 22:05	Admin	
Registro (Ingreso a la Plataforma) de Cursos Académicos	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	02/02/2011 22:05	Admin	
Eliminación de Recursos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	02/02/2011 22:05	Admin	
Incorporación de Recursos al Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	02/02/2011 22:05	Admin	
Historia 5: Gestión de Registro de Estudiante	Plataforma Virtual ESPE-L	Iteración 3	02/03/2011 22:04	Admin	
Historia 7: Gestión de Perfil de Usuario	Plataforma Virtual ESPE-L	Iteración 4	02/03/2011 22:04	Admin	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

Desarrollador: Mónica Llumitasig

Gráfico 4.5. Tareas Pendientes Mónica - Iteración 1

Task Assigned To:	Monica	Task Status:	Pending	Task Count:	5
Task Name	Project Name	Task Group Name	Modified Date	Author	
Búsqueda de un Curso Académico	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	02/02/2011 22:05	Admin	
Visualización de Recursos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	02/02/2011 22:05	Admin	
Modificación de Recursos Académicos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	02/02/2011 22:05	Admin	
Historia 6: Gestión de Calificaciones	Plataforma Virtual ESPE-L	Iteración 3	02/03/2011 22:04	Admin	
Historia 8: Diseño Estético de la Plataforma	Plataforma Virtual ESPE-L	Iteración 4	02/03/2011 22:04	Admin	

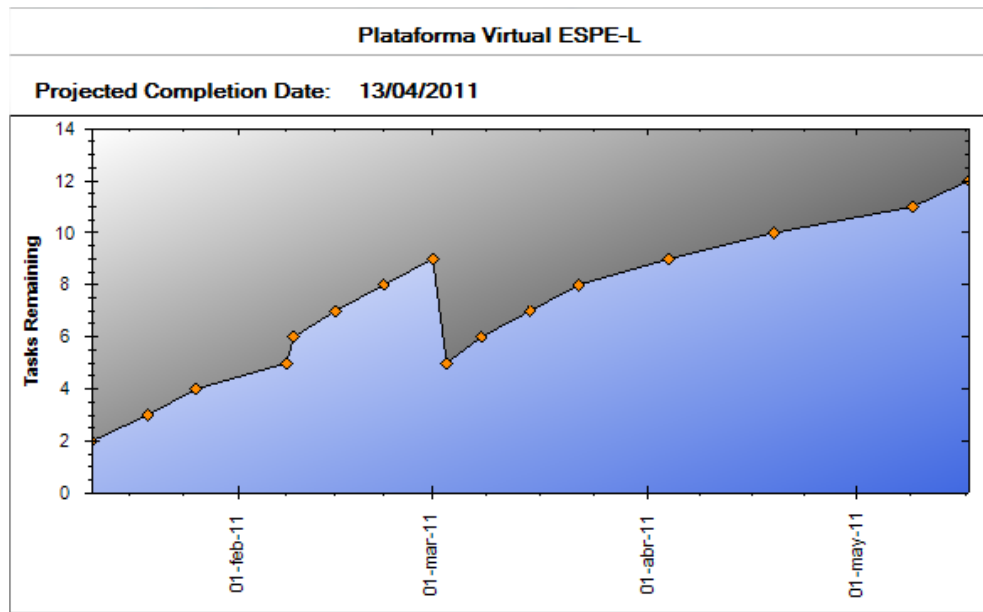
Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.1.2.3.- Burn Down Chart

En el proceso de desarrollo de un proyecto con SCRUM se puede seguir el avance del mismo durante el Sprint por medio de una gráfica llamada Task Burn Down Chart²¹ que nos muestra de que manera se están cumpliendo con los tiempos de entrega establecidos para el Sprint en curso, esta gráfica se presenta en relación a las tareas por cumplir y el tiempo de la última entrega del sistema.

Gráfico 4.6. Burn Down Chart - Iteración 1



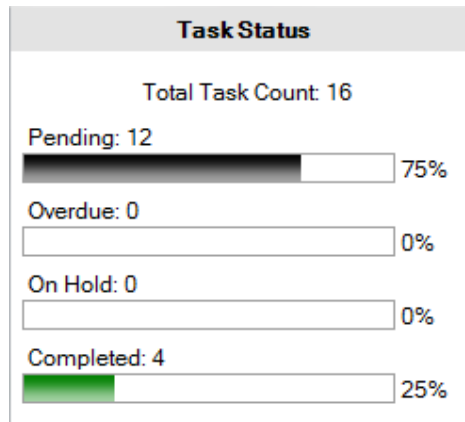
Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

²¹ http://www.dosideas.com/wiki/Gr%C3%A1fico_de_Burn-Down

Así mismo se presenta una gráfica con los porcentajes del estado de las tareas del proceso.

Gráfico 4.7. Estado de Tareas - Iteración 1



Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.2.- ITERACIÓN (SPRINT) 2

4.4.2.1.- Sprint Backlog

Dentro de este Sprint se abarcan las historias de usuario 3 y 4, Gestión de Cursos Académicos y Gestión de Recursos del Curso respectivamente, el Sprint Backlog que se ha obtenido es el siguiente:

Tabla 4.12. Sprint Backlog – Iteración 2

HIST.	TAREA	DELEG	EST	DIAS	8-	15-	22-	29-	36-	43-
				1-7	14	21	28	35	42	49
				HRS PND.	11 76	10 08	840	672	504	336
3	Registro (Ingreso al sistema) de un curso	Marco	Term		X					
3	Búsqueda de un curso académico	Mónica	Term		X					
3	Eliminación de un curso académico	Marco	Term			X				
3	Modificación de un curso académico	Marco	Term				X			
4	Incorporación de recursos al curso	Marco	Term					X		
4	Eliminación de recursos del curso	Marco	Term						X	
4	Modificación de recursos del curso	Mónica	Term							X
4	Visualización de recursos del curso	Mónica	Term							X

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.2.2.-Revisión del Sprint

4.4.2.2.1.- Tareas completadas

Desarrollador: Marco Proaño

Gráfico 4.8. Tareas Completadas Marco - Iteración 2

Task Name	Project Name	Task Group Name	Modified Date	Author
Historia 1: Creación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:00	Admin
Historia 2: Visualización, Modificación, Eliminación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:02	Admin
Incorporación de Recursos al Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	30/03/2011 17:59	Admin
Eliminación de Recursos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	30/03/2011 17:59	Admin
Eliminación de un Curso Académico	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	30/03/2011 17:59	Admin
Registro (Ingreso a la Plataforma) de Cursos Académicos	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	30/03/2011 17:59	Admin
Modificación de un Curso Académico	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	30/03/2011 17:59	Admin

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

Desarrollador: Mónica Llumitasig

Gráfico 4.9. Tareas Completadas Mónica - Iteración 2

Task Assigned To:	Monica	Task Status:	Completed	Task Count:	5
Task Name	Project Name	Task Group Name	Modified Date	Author	
Historia 1: Creación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:01	Admin	
Historia 2: Visualización, Modificación, Eliminación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:02	Admin	
Visualización de Recursos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	30/03/2011 17:59	Admin	
Modificación de Recursos Académicos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	30/03/2011 17:59	Admin	
Búsqueda de un Curso Académico	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	30/03/2011 17:59	Admin	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.2.2.2.- Tareas pendientes

Desarrollador: Marco Proaño

Gráfico 4.10. Tareas Pendientes Marco - Iteración 2

Task Assigned To:	Marco	Task Status:	Pending	Task Count:	2
Task Name	Project Name	Task Group Name	Modified Date	Author	
Historia 5: Gestión de Registro de Estudiante	Plataforma Virtual ESPE-L	Iteración 3	02/03/2011 22:04	Admin	
Historia 7: Gestión de Perfil de Usuario	Plataforma Virtual ESPE-L	Iteración 4	02/03/2011 22:04	Admin	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

Desarrollador: Mónica Llumitasig

Gráfico 4.11. Tareas Pendientes Mónica - Iteración 2

Task Assigned To:	Monica	Task Status:	Pending	Task Count:	4
Task Name	Project Name	Task Group Name	Modified Date	Author	
Visualización de Recursos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	02/02/2011 22:05	Admin	
Modificación de Recursos Académicos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	02/02/2011 22:05	Admin	
Historia 6: Gestión de Calificaciones	Plataforma Virtual ESPE-L	Iteración 3	02/03/2011 22:04	Admin	
Historia 8: Diseño Estético de la Plataforma	Plataforma Virtual ESPE-L	Iteración 4	02/03/2011 22:04	Admin	

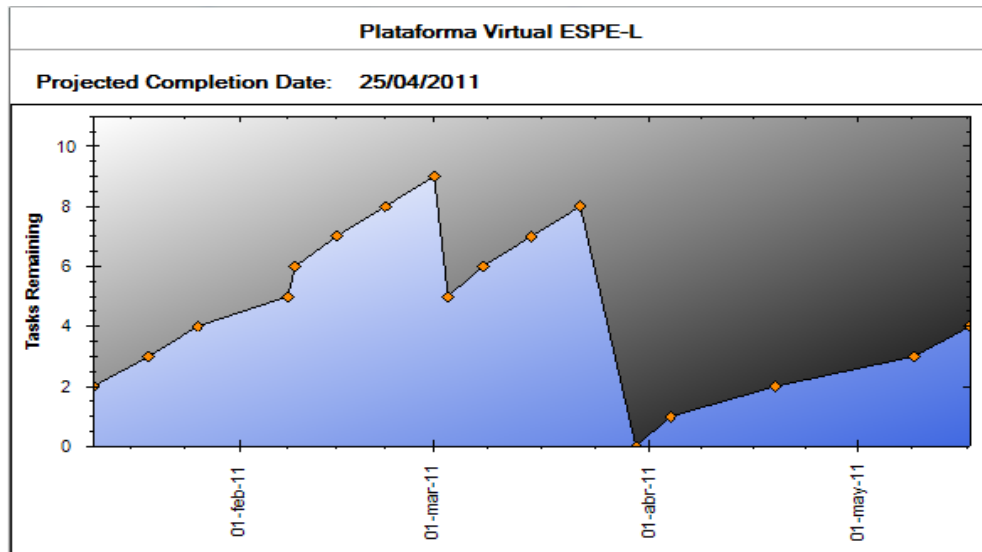
Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.2.2.3.- Burn Down Chart

Se presenta el avance hasta el momento de finalizado el Sprint.

Gráfico 4.12. Burn Down Chart - Iteración 2

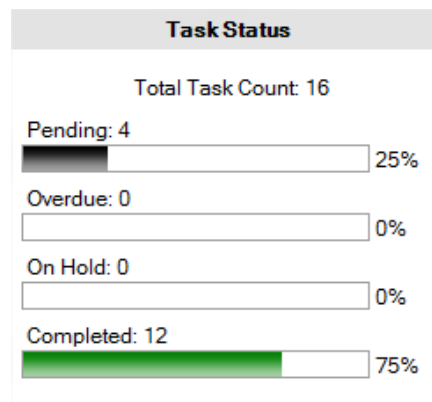


Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

La gráfica con los porcentajes de avance del proyecto es:

Gráfico 4.13. Estado de Tareas - Iteración 2



Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.3.- ITERACIÓN (SPRINT) 3

4.4.3.1.- Sprint Backlog

En esta fase se abarcan las historias de usuario Gestión de Registro de Estudiante y Gestión de Calificaciones, se de talla lo obtenido durante este Sprint.

Tabla 4.13. Sprint Backlog - Iteración 3

HIST.	TAREA	DELEG	ESTAD O	DIAS	1- 7	8- 14	15- 21	22- 28
				HRS PND.	67 2	50 4	336	168
5	Módulo de registro	Marco	Term.		X			
5	Validaciones Id	Marco	Term.		X	X		
5	Validaciones e-mail	Marco	Term.			X		
5	Advertencias	Marco	Term.			X		
6	Diseño módulo de calificaciones	Mónica	Term.				X	
6	Validaciones de registro de calificaciones	Mónica	Term.				X	
6	Modificación de calificaciones	Mónica	Term.					X
6	Visualización de calificaciones	Mónica	Term.					X

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.3.2.-Revisión del Sprint

4.4.3.2.1.- Tareas completadas

Desarrollador: Marco Proaño

Gráfico 4.14. Tareas Completadas Marco - Iteración 3

Task Name	Project Name	Task Group Name	Modified Date	Author
Historia 1: Creación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:00	Admin
Historia 2: Visualización, Modificación, Eliminación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:02	Admin
Incorporación de Recursos al Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	30/03/2011 17:59	Admin
Eliminación de Recursos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	30/03/2011 17:59	Admin
Eliminación de un Curso Académico	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	30/03/2011 17:59	Admin
Registro (Ingreso a la Plataforma) de Cursos Académicos	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	30/03/2011 17:59	Admin
Modificación de un Curso Académico	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	30/03/2011 17:59	Admin
Historia 5: Gestión de Registro de Estudiante	Plataforma Virtual ESPE-L	Iteración 3	04/05/2011 18:37	Admin

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

Desarrollador: Mónica Llumitasig

Gráfico 4.15. Tareas Completadas Mónica - Iteración 3

Task Name	Project Name	Task Group Name	Modified Date	Author
Historia 1: Creación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:01	Admin
Historia 2: Visualización, Modificación, Eliminación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:02	Admin
Visualización de Recursos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	30/03/2011 17:59	Admin
Modificación de Recursos Académicos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	30/03/2011 17:59	Admin
Búsqueda de un Curso Académico	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	30/03/2011 17:59	Admin
Historia 6: Gestión de Calificaciones	Plataforma Virtual ESPE-L	Iteración 3	04/05/2011 18:37	Admin

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.3.2.2.- Tareas pendientes

Desarrollador: Marco Proaño

Gráfico 4.16. Tareas Pendientes Marco - Iteración 3

Task Assigned To:	Marco	Task Status:	Pending	Task Count:	1
Task Name	Project Name	Task Group Name	Modified Date	Author	
Historia 7: Gestión de Perfil de Usuario	Plataforma Virtual ESPE-L	Iteración 4	02/03/2011 22:04	Admin	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

Desarrollador: Mónica Llumitasig

Gráfico 4.17. Tareas Pendientes Mónica - Iteración 3

Task Assigned To:	Monica	Task Status:	Pending	Task Count:	1
Task Name	Project Name	Task Group Name	Modified Date	Author	
Historia 8: Diseño Estético de la Plataforma	Plataforma Virtual ESPE-L	Iteración 4	02/03/2011 22:04	Admin	

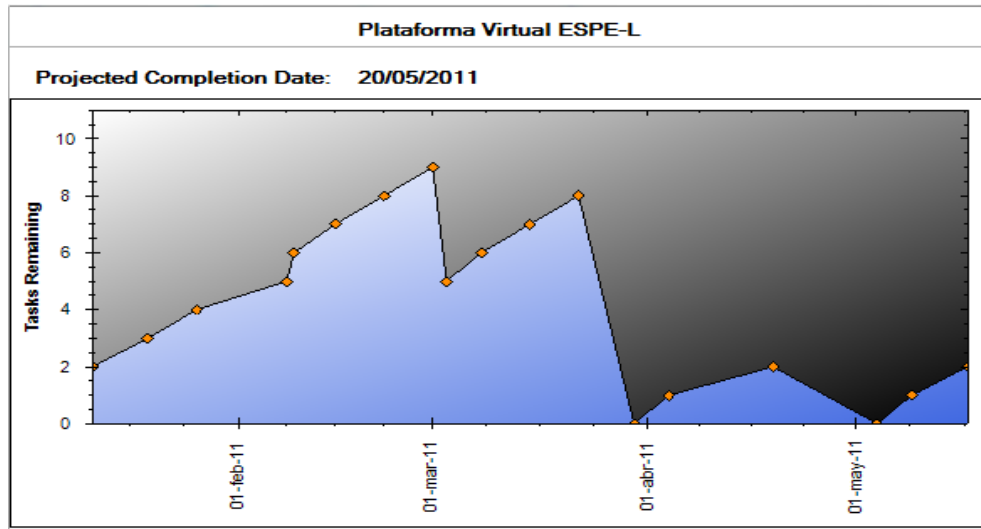
Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.3.2.3.- Burn Down Chart

Se presenta el avance hasta el momento de finalizado el Sprint

Gráfico 4.18. Burn Down Chart - Iteración 3

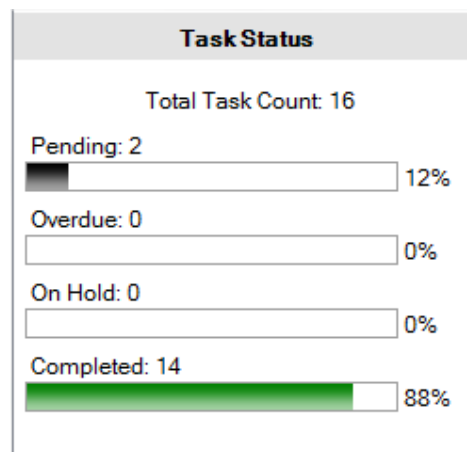


Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

La gráfica con los porcentajes de avance del proyecto es:

Gráfico 4.19. Estado de Tareas - Iteración 3



Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.4.- ITERACIÓN (SPRINT) 4

4.4.4.1.- Sprint Backlog

En esta fase se abarcan las historias de usuario de Gestión de Perfil de Usuario y de Diseño Estético de la Plataforma, se detalla lo obtenido durante este último Sprint.

Tabla 4.14. Sprint Backlog - Iteración 4

HIST	TAREA	DELEGADO	ESTADO	DIAS	1-7	8-14
				HRS PND.	67 2	504
7	Módulo de perfil de usuario	Marco	Term.		X	
7	Editar información usuario	Marco	Term.		X	X
7	Visualizar información usuario	Marco	Term.			X
7	Módulo mensajes	Marco	Term.		x	X
7	Módulo Blog	Mónica	Term.		x	
8	Elección de plantilla	Mónica	Term.		x	
8	Diseño visualizaciones adicionales	Mónica	Term.			x
8	Visualización de entorno de plataforma	Mónica	Term.		x	x

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.4.2.-Revisión del Sprint

4.4.4.2.1.- Tareas completadas

Desarrollador: Marco Proaño

Gráfico 4.20. Tareas Completadas Marco - Iteración 4

Task Assigned To:	Marco	Task Status:	Completed	Task Count:	9
Task Name	Project Name	Task Group Name	Modified Date	Author	
Historia 1: Creación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:00	Admin	
Historia 2: Visualización, Modificación, Eliminación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:02	Admin	
Incorporación de Recursos al Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	30/03/2011 17:59	Admin	
Eliminación de Recursos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	30/03/2011 17:59	Admin	
Eliminación de un Curso Académico	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	30/03/2011 17:59	Admin	
Registro (Ingreso a la Plataforma) de Cursos Académicos	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	30/03/2011 17:59	Admin	
Modificación de un Curso Académico	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	30/03/2011 17:59	Admin	
Historia 5: Gestión de Registro de Estudiante	Plataforma Virtual ESPE-L	Iteración 3	04/05/2011 18:37	Admin	
Historia 7: Gestión de Perfil de Usuario	Plataforma Virtual ESPE-L	Iteración 4	25/05/2011 18:48	Admin	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

Desarrollador: Mónica Llumitasig

Gráfico 4.21. Tareas Completadas Mónica - Iteración 4

Task Assigned To:	Monica	Task Status:	Completed	Task Count:	7
Task Name	Project Name	Task Group Name	Modified Date	Author	
Historia 1: Creación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:01	Admin	
Historia 2: Visualización, Modificación, Eliminación de Carreras o Programas Académicos	Plataforma Virtual ESPE-L	Iteración 1	02/03/2011 22:02	Admin	
Visualización de Recursos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	30/03/2011 17:59	Admin	
Modificación de Recursos Académicos del Curso	Plataforma Virtual ESPE-L	Historia 4: Gestión de Recursos del Curso	30/03/2011 17:59	Admin	
Búsqueda de un Curso Académico	Plataforma Virtual ESPE-L	Historia 3: Gestión de Cursos Académicos	30/03/2011 17:59	Admin	
Historia 6: Gestión de Calificaciones	Plataforma Virtual ESPE-L	Iteración 3	04/05/2011 18:37	Admin	
Historia 8: Diseño Estético de la Plataforma	Plataforma Virtual ESPE-L	Iteración 4	25/05/2011 18:48	Admin	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

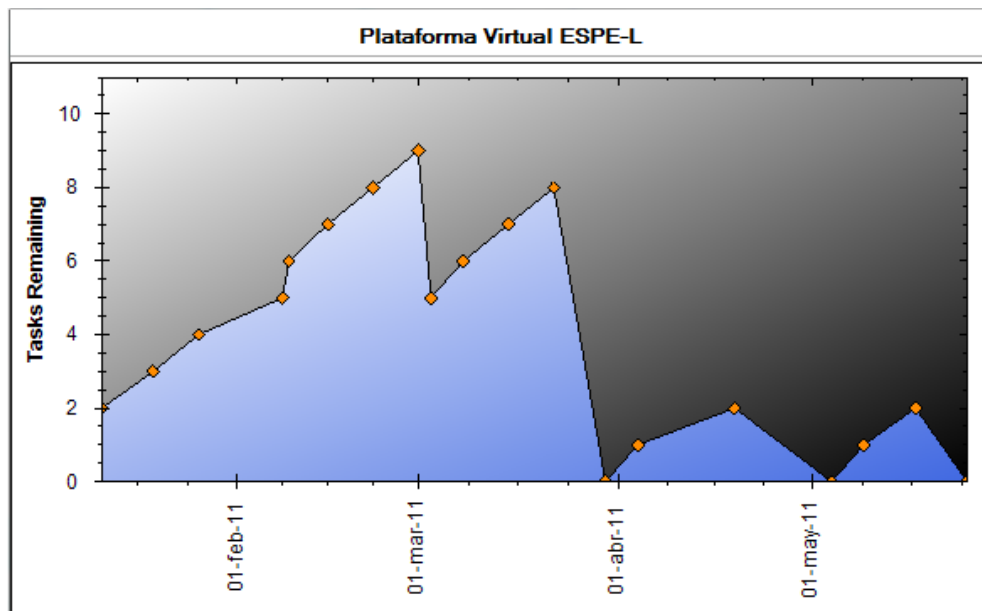
4.4.4.2.2.- Tareas pendientes

Al ser este el último Sprint y al haber llevado a cabalidad el desarrollo de nuestra plataforma, no existen tareas sin culminar en el proceso de desarrollo, aunque cabe recalcar que al ser este un prototipo y un sistema en crecimiento constante las tareas seguirán apareciendo de acuerdo con el incremento de las necesidades del Product Owner a lo largo de la vida útil del sistema, estas deberán ser agregadas al final Product Backlog para ser lanzadas en una nueva versión.

4.4.4.2.3.- Burn Down Chart

Se presenta el avance hasta el momento de finalizado el Sprint

Gráfico 4.22. Burn Down Chart - Iteración 4

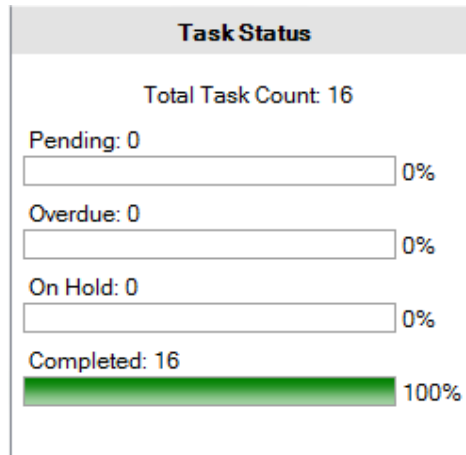


Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

La gráfica con los porcentajes de avance del proyecto es:

Gráfico 4.23. Estado de Tareas - Iteración 4



Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.4.4.3.-Product Backlog Final

Al finalizar el proceso de desarrollo nos encontramos con el listado de tareas basadas en las Historias de Usuario pero ya delimitadas, asignadas y terminadas.

Gráfico 4.24.Gráfico 3.24. Product Backlog Final

	Project / Task Group / Task Name	Status	Assigned To	Start Date	Due Date	Completed Status	Modified By	Author	Archive
	Plataforma Virtual ESPE-L	Complete		10/01/2011	23/05/2011 0:00	25/05/2011 18:48	Marco	Admin	
	Iteración 1	Complete		10/01/2011	31/01/2011 7:51	02/03/2011 22:02	Marco	Admin	
★	Historia 1: Creación de Carreras o Programas Académicos	Completed	Monica	10/01/2011 23:	17/01/2011 7:43	02/03/2011 22:01	Marco	Admin	
★	Historia 1: Creación de Carreras o Programas Académicos	Completed	Marco	10/01/2011 23:	24/01/2011 7:43	02/03/2011 22:00	Marco	Admin	
★	Historia 2: Visualización, Modificación, Eliminación de Carrera	Completed	Monica	25/01/2011 23:	31/01/2011 7:51	02/03/2011 22:02	Marco	Admin	
★	Historia 2: Visualización, Modificación, Eliminación de Carrera	Completed	Marco	18/01/2011 23:	31/01/2011 7:51	02/03/2011 22:02	Marco	Admin	
	Iteración 2	Complete		07/02/2011	28/03/2011 8:19	30/03/2011 17:59	Marco	Admin	
★	Historia 3: Gestión de Cursos Académicos	Complete		07/02/2011	28/02/2011 8:08	30/03/2011 17:59	Marco	Admin	
	Búsqueda de un Curso Académico	Completed	Monica	09/02/2011 0:1	14/02/2011 0:10	30/03/2011 17:59	Marco	Admin	
	Eliminación de un Curso Académico	Completed	Marco	22/02/2011 0:0	28/02/2011 8:08	30/03/2011 17:59	Marco	Admin	
	Modificación de un Curso Académico	Completed	Marco	15/02/2011 0:0	21/02/2011 8:06	30/03/2011 17:59	Marco	Admin	
	Registro (Ingreso a la Plataforma) de Cursos Académicos	Completed	Marco	07/02/2011 23:	14/02/2011 23:58	30/03/2011 17:59	Marco	Admin	
★	Historia 4: Gestión de Recursos del Curso	Complete		01/03/2011	28/03/2011 8:19	30/03/2011 17:59	Marco	Admin	
	Eliminación de Recursos del Curso	Completed	Marco	15/03/2011 0:1	21/03/2011 8:17	30/03/2011 17:59	Marco	Admin	
	Incorporación de Recursos al Curso	Completed	Marco	01/03/2011 0:1	07/03/2011 0:13	30/03/2011 17:59	Marco	Admin	
	Modificación de Recursos Académicos del Curso	Completed	Monica	08/03/2011 0:1	14/03/2011 0:15	30/03/2011 17:59	Marco	Admin	
	Visualización de Recursos del Curso	Completed	Monica	22/03/2011 0:1	28/03/2011 8:19	30/03/2011 17:59	Marco	Admin	
	Iteración 3	Complete		04/04/2011	02/05/2011 0:24	04/05/2011 18:37	Marco	Admin	
★	Historia 5: Gestión de Registro de Estudiante	Completed	Marco	04/04/2011 0:2	18/04/2011 0:21	04/05/2011 18:37	Marco	Admin	
★	Historia 6: Gestión de Calificaciones	Completed	Monica	19/04/2011 0:2	02/05/2011 0:24	04/05/2011 18:37	Marco	Admin	
	Iteración 4	Complete		09/05/2011	23/05/2011 8:28	25/05/2011 18:48	Marco	Admin	
	Historia 7: Gestión de Perfil de Usuario	Completed	Marco	09/05/2011 0:2	16/05/2011 0:26	25/05/2011 18:48	Marco	Admin	
	Historia 8: Diseño Estético de la Plataforma	Completed	Monica	17/05/2011 0:2	23/05/2011 8:28	25/05/2011 18:48	Marco	Admin	

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

4.5.- PRUEBAS

De acuerdo a lo estipulado en el capítulo 2, en la explicación de las metodologías Extreme Programming y SCRUM se planteará un plan de pruebas para los casos más relevantes en el uso y manipulación de nuestro prototipo.

4.5.1.- PLAN DE PRUEBAS

- **Prueba de registro de un nuevo usuario**

En esta prueba realizamos el registro de un nuevo usuario dentro del sistema mediante el formulario de registro.

Tabla 4.15.Tabla 3.15 Prueba de registro de un nuevo usuario

Caso de Prueba	Registro de usuario
Entrada	Ingreso de Datos
	Nombre de Usuario: Estudiante
	Contraseña: *****
	Dirección de correo: estudiante@hotmail.com
	Correo (de nuevo): estudiante@hotmail.com
	Apellido: Estudiante
	Nombre: Pruebas
	Ciudad: Quito
	País: Ecuador
Resultado Esperado	El sistema debe permitir el registro del nuevo usuario
Condiciones	El nombre de usuario debe ser único
	El mail de usuario debe ser único
Observación	El usuario no puede registrarse varias veces, se valida el nombre de usuario y el correo electrónico, de existir el nombre de usuario o el mail dentro del sistema, este mostrara un mensaje informativo.

Procedimiento	Ingresar a la plataforma virtual
	Hacer click sobre "Comience ahora creando una cuenta"
	Ingresar los datos obligatorios (Nombre de usuario, Contraseña, Dirección de correo, Correo, Apellido, Nombre, Ciudad, País).
	Ingresar el reCaptcha.
	Hacer click en "Crear cuenta"
Resultado Real	El sistema carga la página de inicio de sesión para permitir el ingreso al nuevo usuario creado.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de eliminación de usuario**

Describiremos como el administrador del sistema puede dar de baja a un usuario registrado dentro del sistema.

Tabla 4.16. Prueba de Eliminación de Usuarios

Caso de Prueba	Eliminación de usuarios (Administrador)
Entrada	Ingreso de Datos
	Nombre de Usuario: Adminrpruebas
	Contraseña: *****
Resultado Esperado	El sistema elimina el usuario deseado.
Condiciones	El usuario debe ser administrador para realizar la eliminación.
Observación	Se puede realizar solo dando click sobre el botón de eliminar.
Procedimiento	Ingresar a la plataforma virtual
	Ingresar los campos: Nombre de Usuario, Password y captcha
	Oprimir el botón "Entrar"

Hacer click sobre "Cuentas" en el menú de

	Administración del sitio.
	Hacer click sobre "Hojea lista de usuarios"
	Localizar el usuario que se desea eliminar (Ej. Estudiante Pruebas)
	Hacer click sobre "Borrar" al costado derecho del nombre del estudiante en cuestión
	Hacer click sobre el botón "Si"
Resultado Real	El sistema elimina el usuario y despliega la página anterior con la lista de usuarios existentes dentro del sistema.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de inicio de sesión en el sistema**

En esta prueba se procede con el ingreso al sistema de un usuario previamente registrado en el mismo.

Tabla 4.17. Prueba de Inicio de Sesión

Caso de Prueba	Inicio de sesión de usuario
Entrada	Ingreso de Datos
	Nombre de Usuario: Estudiante
	Contraseña: *****
Resultado Esperado	El sistema despliega el perfil del usuario correspondiente al usuario que ha ingresado.
Condiciones	El usuario debe existir dentro del sistema, y deben coincidir, el nombre de usuario, la contraseña y el código captcha
Observación	En caso de error, el sistema regresará a la página de inicio y mostrara un mensaje informativo
Procedimiento	Ingresar a la plataforma virtual
	Ingresar los campos: Nombre de Usuario, Password y captcha

	Oprimir el botón "Entrar"
Resultado Real	El sistema despliega el menú administrativo correspondiente al usuario que ha ingresado

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de Creación de Carreras o Programas Académicos**

Esta prueba consiste en la creación de una carrera o programa académico dentro del sistema para permitir la interacción de usuarios de dicha carrera.

Tabla 4.18. Prueba de Creación de Carrera o Programa Académico

Caso de Prueba	Creación de carrera o Programa Académico
Entrada	Ingreso de Datos
	Categoría Padre: "Top"
	Nombre de categoría: "Postgrado"
	Descripción: Maestrías que se dictan dentro de la institución.
Resultado Esperado	El sistema debe guardar los datos de la nueva carrera creada previa a la verificación de los mismos y trasladar al listado de carreras.
Condiciones	Se debe ingresar obligatoriamente el nombre de la categoría
Observación	En caso de error (no ingresada la categoría), el sistema regresará a la página de inicio y mostrara un mensaje informativo de que el campo es obligatorio
Procedimiento	Ingresar a la plataforma virtual (Administrador)
	Ingresar a la sección de cursos (Agregar/Editar Cursos)
	Oprimir el botón "Agregar categoría" (Tipos/Carreras)

Ingresar los campos: Categoría padre, Nombre de

	categoría y Descripción. Oprimir el botón "Crear categoría"
Resultado Real	El sistema regresa a la pantalla anterior de edición de Carreras con la carrera agregada anteriormente.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de listado de carreras o programas académicos**

Esta prueba consiste en la presentación de una lista de carreras o programas académicos registrados previamente en el sistema.

Tabla 4.19. Prueba de Listado de Carreras

Caso de Prueba	Listado de Carreras
Entrada	Ingreso de Datos
	Nombre de Usuario: 1804013082
	Contraseña: *****
Resultado Esperado	El sistema despliega la pantalla de bienvenida con el listado de carreras existentes en el sistema.
Condiciones	El usuario debe existir dentro del sistema, y deben coincidir, el nombre de usuario, la contraseña y el código captcha
Observación	En caso de error, el sistema regresará a la página de inicio y mostrara un mensaje informativo
Procedimiento	Ingresar a la plataforma virtual
	Ingresar los campos: Nombre de Usuario, Password y captcha
	Oprimir el botón "Entrar"
Resultado Real	El sistema despliega el menú administrativo correspondiente al usuario que ha ingresado.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de Visualización de Carreras o Programas Académicos**

En esta prueba describiremos el proceso para revisar la lista de cursos e información básica de una carrera determinada.

Tabla 4.20. Prueba de Visualización de Carreras

Caso de Prueba	Visualización de carreras
Entrada	Ingreso de Datos
	Nombre de Usuario: 1804013082
	Contraseña: *****
Resultado Esperado	El sistema despliega la lista de cursos e información básica pertenecientes a la carrera seleccionada.
Condiciones	El usuario debe existir dentro del sistema, y deben coincidir, el nombre de usuario, la contraseña y el código captcha.
Observación	Se puede seleccionar dando click sobre el nombre de la carrera.
Procedimiento	Ingresar a la plataforma virtual
	Ingresar los campos: Nombre de Usuario, Password y captcha
	Oprimir el botón "Entrar"
	Hacer click sobre la carrera Ej. (Ciencias Exactas)
	Visualizar la lista de cursos e información de la carrera.
Resultado Real	El sistema despliega en la página la información de Ciencias Exactas así como la lista de cursos disponibles en esta carrera.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de Modificación de Carreras o Programas Académicos**

En esta prueba se realiza la modificación de la información de una carrera registrada en el sistema.

Tabla 4.21. Prueba de Modificación de Carreras

Caso de Prueba	Modificación de carreras (Administrador)
Entrada	Ingreso de Datos
	Nombre de Usuario: Adminrpruebas
	Contraseña: *****
Resultado Esperado	El sistema despliega la lista de campos modificables de la carrera seleccionada.
Condiciones	El usuario debe ser administrador para realizar la modificación de la carrera.
Observación	Se puede seleccionar dando click sobre el nombre de la carrera.
Procedimiento	Ingresar a la plataforma virtual
	Ingresar los campos: Nombre de Usuario, Password y captcha
	Oprimir el botón "Entrar"
	Hacer click sobre "Cursos" en el menú de Administración del sitio.
	Hacer click sobre "Agregar/Editar Cursos"
	Hacer click sobre el ícono de editar al costado derecho de la carrera en cuestión (Ej. Ciencias Exactas)
	Modificar los campos que se deseen (Ej. Categoría Padre, Nombre, Descripción)
	Hacer click sobre el botón "Guardar Cambios"
Resultado Real	El sistema despliega los campos editables correspondientes a la carrera elegida.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de Eliminación de Carreras o Programas Académicos**

Se realiza la eliminación de una carrera por parte del administrador del sistema.

Tabla 4.22. Prueba de Eliminación de Carreras

Caso de Prueba	Eliminación de carreras (Administrador)
Entrada	Ingreso de Datos
	Nombre de Usuario: Adminrpruebas
	Contraseña: *****
Resultado Esperado	El sistema despliega la lista de carreras en la página principal pero sin la carrera eliminada.
Condiciones	El usuario debe ser administrador para realizar la eliminación de la carrera.
Observación	Se puede seleccionar dando click sobre el nombre de la carrera.
Procedimiento	Ingresar a la plataforma virtual
	Ingresar los campos: Nombre de Usuario, Password y captcha
	Oprimir el botón "Entrar"
	Hacer click sobre "Cursos" en el menú de Administración del sitio.
	Hacer click sobre "Agregar/Editar Cursos"
	Hacer click sobre el ícono de eliminar al costado derecho de la carrera en cuestión (Ej. Ciencias Exactas)
	Hacer click sobre el botón "Borrar"
	Hacer click sobre el botón "Continuar"
Resultado Real	El sistema despliega el menú administrativo correspondiente a "Agregar/Editar Cursos" con la lista de cursos existentes.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de creación de cursos en una carrera o programa académico**

Se realiza la creación de un curso nuevo dentro de una carrera existente dentro del sistema.

Tabla 4.23. Prueba de creación de cursos en una carrera o programa académico

Caso de Prueba	Creación de Cursos
Entrada	Ingreso de Datos
	Categoría: "POSTGRADO / Maestría en Ingeniería de Software"
	Nombre Completo: "Curso de Prueba"
	Nombre Corto: "PCDP"
	Número ID del curso: ""
	Resumen: Este es un curso creado para una prueba.
Resultado Esperado	El sistema debe guardar los datos del nuevo curso creado previo a la verificación de los mismos y trasladar al listado de carreras.
Condiciones	El usuario debe ser administrador para poder realizar la creación del curso
	Se debe ingresar obligatoriamente el nombre de curso y el nombre abreviado
Observación	En caso de error, el sistema regresará a la página de creación de curso y mostrara un mensaje informativo de que el campo es obligatorio o ya se ha ingresado uno igual
Procedimiento	Ingresar a la plataforma virtual (Administrador)
	Ingresar a la sección de cursos (Agregar/Editar Cursos)
	Oprimir el botón "Agregar curso" (Tipos/Carreras)
	Ingresar los campos: Categoría, Nombre completo.

	Oprimir el botón "Crear categoría"
Resultado Real	El sistema regresa a la pantalla anterior de edición de Carreras con la carrera agregada anteriormente.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de dada de baja a un curso que pertenece a una carrera o programa académico**

Se demuestra la dada de baja de un curso registrado dentro de una carrera o programa académico existente dentro del sistema.

Tabla 4.24. Prueba de dada de baja a los cursos que pertenecen a una carrera o programa académico

Caso de Prueba	Eliminación de cursos (Administrador)
Entrada	Ingreso de Datos
	Nombre de Usuario: Adminrpruebas
	Contraseña: *****
Resultado Esperado	El sistema elimina el curso deseado y despliega la lista de cursos pertenecientes a la carrera.
Condiciones	El usuario debe ser administrador para realizar la eliminación del curso.
Observación	Se puede realizar solo dando click sobre el botón de eliminar.
Procedimiento	Ingresar a la plataforma virtual
	Ingresar los campos: Nombre de Usuario, Password y captcha
	Oprimir el botón "Entrar"
	Hacer click sobre "Cursos" en el menú de Administración del sitio.
	Hacer click sobre "Agregar/Editar Cursos"

	Hacer click sobre la carrera que contenga el curso (Ej. Ciencias Exactas)
	Hacer click sobre el ícono de eliminar al costado derecho del curso en cuestión (Ej. Ciencias Exactas)
	Hacer click sobre el botón "Si"
	Hacer click sobre el botón "Continuar"
Resultado Real	El sistema elimina el curso y despliega la página anterior con la lista de cursos existentes dentro de la carrera elegida.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de modificación de un curso perteneciente a una carrera o programa académico**

En esta prueba se realiza la modificación de la información básica o de parámetros de configuración de un curso perteneciente a una carrera.

Tabla 4.25. Prueba de modificación de un curso perteneciente a una carrera o programa académico

Caso de Prueba	Modificación de Cursos (Administrador)
Entrada	Ingreso de Datos
	Nombre de Usuario: Adminrpruebas
	Contraseña: *****
Resultado Esperado	El sistema despliega la lista de campos modificables de la carrera seleccionada.
Condiciones	El usuario debe ser administrador para realizar la modificación del curso.
Observación	La modificación (actualización) se realiza por medio del botón configuración.
Procedimiento	Ingresar a la plataforma virtual

	Ingresar los campos: Nombre de Usuario, Password y captcha
	Oprimir el botón "Entrar"
	Hacer click sobre "Cursos" en el menú de Administración del sitio.
	Hacer click sobre "Agregar/Editar Cursos"
	Hacer click sobre la carrera donde se encuentra alojado el curso a modificar Ej. (Maestría en Ingeniería del Software).
	Hacer click sobre el ícono de Configuración al costado derecho del curso en cuestión (Ej. Gestión del Desarrollo del Proceso de Software)
	Modificar los campos que se deseen (Ej. Categoría, Nombre Completo, Nombre Corto, Resumen, etc.)
	Hacer click sobre el botón "Guardar Cambios"
Resultado Real	El sistema despliega el menú administrativo correspondiente a "Agregar/Editar Cursos" con la lista de cursos existentes.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de asignación de roles globales**

En esta prueba se procede con la asignación de los roles de forma global a usuarios registrados previamente en el sistema.

Tabla 4.26. Prueba de Asignación de Roles Globales

Caso de Prueba	Asignación de Roles (Globales)
Entrada	Ingreso de Datos
	Nombre de Usuario: Adminpruebas
	Contraseña: *****
Resultado Esperado	El sistema debe guardar los cambios de asignación de roles globales.

Condiciones	El usuario debe ser administrador para realizar los cambios
Observación	Los roles que se asignen serán de tipo global, es decir se aplicarán para todo el sitio.
Procedimiento	Ingresar a la plataforma virtual (Administrador)
	Ingresar a la sección de usuarios (Permisos)
	Ingresar a la sección "Asignar roles globales"
	Hacer click en el nombre del rol que se desea asignar Ej.("Administrador")
	Ingresar los campos: Categoría padre, Nombre de categoría y Descripción.
	Pasar de la lista de usuarios registrados a la lista del rol que estemos asignando
	Hacer click en "Asignar roles en Núcleo del Sistema"
Resultado Real	El sistema regresa a la pantalla anterior (Asignar roles globales) donde se aprecia la tabla de roles con los nuevos usuarios a los q se les asigno roles globales

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de asignación de roles (Curso)**

En esta prueba se realiza la asignación de roles (profesor, estudiante) dentro de un curso determinado por parte del administrador, cabe recalcar que dependiendo de la configuración del curso creado la asignación del rol de estudiante se realiza de forma automática por medio de un password proporcionado por el profesor del curso al momento del registro dentro del mismo.

Tabla 4.27. Prueba de Asignación de Roles (Curso)

Caso de Prueba	Asignación de Roles (Curso)
Entrada	Ingreso de Datos
	Nombre de Usuario: Adminpruebas
	Contraseña: *****
Resultado Esperado	El sistema debe guardar los cambios de asignación de roles.
Condiciones	El usuario debe ser administrador para realizar los cambios
Observación	Los roles que se asignen serán de tipo local, es decir se aplicarán solamente para el curso seleccionado.
Procedimiento	Ingresar a la plataforma virtual (Administrador)
	Hacer click sobre "Cursos" en el menú de Administración del sitio.
	Hacer click sobre "Agregar/Editar Cursos"
	Hacer click sobre la carrera donde se encuentra alojado el curso a modificar Ej. (Maestría en Ingeniería del Software).
	Hacer click sobre el ícono de Asignar roles al costado derecho del curso en cuestión (Ej. Gestión del Desarrollo del Proceso de Software)
	Pasar de la lista de usuarios registrados a la lista del rol que estemos asignando
	Hacer click en "Asignar roles en Curso: PIMS"
Resultado Real	El sistema regresa a la pantalla anterior (Asignar roles en Curso: PIMS) donde se aprecia la tabla de roles con los nuevos usuarios a los q se les asigno los roles

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de creación de sala de Chat**

Se procede con la creación de una sala de Chat por parte del profesor del curso, este estará disponible solamente desde la fecha en que el profesor creyere conveniente.

Tabla 4.28. Prueba de Creación de Sala de Chat

Caso de Prueba	Creación de sala de chat (Profesor)
Entrada	Ingreso de Datos
	Nombre de Usuario: Profesor
	Contraseña: *****
Resultado Esperado	El sistema debe mostrar el modulo con la sala de chat creada.
Condiciones	El usuario debe estar registrado como profesor del curso
Observación	La sala de chat estará abierta durante todo el periodo académico, y desde la fecha en que el profesor decida publicarla.
Procedimiento	Ingresar a la plataforma virtual (Profesor)
	Ingresar al curso donde se desee crear la sala de chat Ej.(Gestión del Desarrollo del Proceso de Software)
	En la sección de "Administración", hacer click sobre "Activar edición"
	En el modulo que se desee crear la sala seleccionar dentro de la lista de "Agregar actividad..." la opción de "Chat"
	Ingresar datos como "Nombre de la sala", "Texto introductorio" y los ajustes opcionales.
	Hacer click en "Guardar cambios y regresar al curso" o en "Guardar cambios y mostrar"

Resultado Real	Click sobre "Guardar cambios y regresar al curso": El sistema regresa a la página principal del curso con la sala de chat creada.
	Click sobre "Guardar cambios y mostrar": El sistema hace una dirección a la página para ingresar a la sala de Chat.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de utilización de sala de Chat**

En esta prueba se realiza la interacción del usuario (Estudiante) dentro de la sala de chat creada con anterioridad.

Tabla 4.29. Prueba de Utilización de Sala de Chat

Caso de Prueba	Ingreso e interacción en sala de chat
Entrada	Ingreso de Datos
	Nombre de Usuario: Estudiante
	Contraseña: *****
Resultado Esperado	El sistema debe permitir el ingreso e interacción dentro de la sala de Chat.
Condiciones	El usuario debe estar registrado dentro del curso para poder interactuar dentro de la sala de chat
	El usuario puede ser profesor del curso
Observación	La sala de chat estará abierta desde la fecha configurada por el profesor.
Procedimiento	Ingresar a la plataforma virtual
	Ingresar al curso donde se ha abierto una sala de chat Ej.(Gestión del Desarrollo del Proceso de Software)
	Hacer click sobre el nombre de la sala de chat Ej.(Chat MIS-001)
	Hacer click sobre "Entrar a la sala"

	Interactuar dentro de la sala
Resultado Real	El sistema abre una página (pop-up) con la sala de chat y los mensajes de los demás usuarios y la lista de usuarios conectados

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de creación de foros**

Se realiza la creación de un foro que permitirá la interacción de los usuarios durante un periodo de tiempo determinado por el profesor.

Tabla 4.30. Prueba de Creación de Foros

Caso de Prueba	Creación de Foro (Profesor)
Entrada	Ingreso de Datos
	Nombre de Usuario: Profesor
	Contraseña: *****
Resultado Esperado	El sistema debe mostrar el modulo con el foro creado.
Condiciones	El usuario debe estar registrado como profesor del curso
Observación	El foro estará disponible desde su fecha de creación, de manera indefinida o por el periodo de días que el profesor lo considere.
Procedimiento	Ingresar a la plataforma virtual (Profesor)
	Ingresar al curso donde se desee crear el foro Ej.(Gestión del Desarrollo del Proceso de Software)
	En la sección de "Administración", hacer click sobre "Activar edición"
	En el modulo que se desee crear la sala seleccionar dentro de la lista de "Agregar actividad..." la opción de "Foro"

	Ingresar datos como "Nombre del foro", "Introducción" y los ajustes opcionales.
	Hacer click en "Guardar cambios y regresar al curso" o en "Guardar cambios y mostrar"
Resultado Real	Click sobre "Guardar cambios y regresar al curso": El sistema regresa a la pagina principal del curso con el foro creado dentro del modulo seleccionado.
	Click sobre "Guardar cambios y mostrar": El sistema realiza una redirección a la página del foro creado para la interacción.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de ingreso e interacción en foros**

En esta prueba se demuestra el proceso para la interacción del usuario (estudiante) dentro de un foro creado anteriormente.

Tabla 4.31. Prueba de Ingreso e Interacción en Foros

Caso de Prueba	Ingreso e interacción en foros
Entrada	Ingreso de Datos
	Nombre de Usuario: Estudiante
	Contraseña: *****
Resultado Esperado	El sistema debe permitir el ingreso e interacción dentro del foro.
Condiciones	El usuario debe estar registrado dentro del curso para poder interactuar dentro del foro.
	El usuario puede ser profesor del curso
Observación	El foro permite la subida de archivos.
Procedimiento	Ingresar a la plataforma virtual

Ingresar al curso donde se ha abierto un foro Ej.(Gestión

	del Desarrollo del Proceso de Software)
	Hacer click sobre el nombre del foro Ej.(FAQ "El arte de la ingeniería de Software")
	Hacer click sobre "Responder"
	Interactuar enviando un mensaje de respuesta al tema propuesto o subiendo un archivo en la respuesta.
Resultado Real	El sistema abre una página donde esta ubicado el foro y aparecen los mensajes de los demás usuarios y el botón para responder.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de carga de material de consulta (Profesor)**

Se demuestra el proceso para la carga de material de consulta para estudiantes por parte del profesor del curso.

Tabla 4.32. Prueba de Carga de Material de Consulta (Profesor)

Caso de Prueba	Carga de material de consulta (Profesor)
Entrada	Ingreso de Datos
	Nombre de Usuario: Profesor
	Contraseña: *****
Resultado Esperado	El sistema debe cargar el archivo de consulta.
Condiciones	El usuario debe estar registrado como profesor del curso
Observación	El archivo se subirá al sistema directamente en el modulo seleccionado.
Procedimiento	Ingresar a la plataforma virtual (Profesor)
	Ingresar al curso donde se desee cargar el archivo de consulta Ej.(Gestión del Desarrollo del Proceso de Software)

	En la sección de "Administración", hacer click sobre "Activar edición"
	En el modulo que se desee crear la sala seleccionar dentro de la lista de "Agregar recurso..." la opción de "Enlazar un archivo o una Web"
	Ingresar datos como "Nombre" y "Ubicación".
	Hacer click en "Elija o suba un archivo ..."
	Hacer click en "Subir un archivo"
	Hacer click en "Seleccionar un archivo"
	Seleccionar el archivo que se desea subir a la plataforma.
	Hacer click en "Subir este archivo"
	Hacer click sobre "Elegir" al costado derecho del nombre del archivo a subir.
	Hacer click en "Guardar cambios y regresar al curso" o en "Guardar cambios y mostrar"
Resultado Real	Click sobre "Guardar cambios y regresar al curso": El sistema regresa a la pagina principal del curso con el archivo de consulta subido dentro del modulo seleccionado.
	Click sobre "Guardar cambios y mostrar": El sistema descarga el archivo que se ha subido con anterioridad.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de creación de una tarea (Profesor)**

En esta prueba se realiza la creación de una tarea (calificada) por parte del profesor del curso.

Tabla 4.33. Prueba de Creación de una Tarea (Profesor)

Caso de Prueba	Creación de Tarea (Profesor)
Entrada	Ingreso de Datos
	Nombre de Usuario: Profesor
	Contraseña: *****
Resultado Esperado	El sistema debe mostrar el modulo con la tarea creada.
Condiciones	El usuario debe estar registrado como profesor del curso
Observación	La tarea estará disponible desde su fecha de creación, de manera indefinida o por el periodo de días que el profesor lo considere.
Procedimiento	Ingresar a la plataforma virtual (Profesor)
	Ingresar al curso donde se desee crear la tarea Ej.(Gestión del Desarrollo del Proceso de Software)
	En la sección de "Administración", hacer click sobre "Activar edición"
	En el modulo que se desee crear la sala seleccionar dentro de la lista de "Agregar actividad..." y debajo de "Tarea" la opción de "Subir un solo archivo"
	Ingresar datos como "Nombre de la tarea", "Descripción" y los ajustes opcionales.
	Hacer click en "Guardar cambios y regresar al curso" o en "Guardar cambios y mostrar"

Resultado Real	Click sobre "Guardar cambios y regresar al curso": El sistema regresa a la pagina principal del curso con la tarea creada dentro del modulo seleccionado.
	Click sobre "Guardar cambios y mostrar": El sistema redirecciona a la página de la tarea creada con su vista previa.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de interacción (consulta, resolución) en tareas**

En esta prueba se realiza la interacción ya sea esta la resolución, consulta o subida de archivos dentro de una tarea por parte de un estudiante registrado en un curso determinado.

Tabla 4.34. Prueba de Interacción (Consulta, Resolución) en Tareas

Caso de Prueba	Interacción en Tareas (Estudiante)
Entrada	Ingreso de Datos
	Nombre de Usuario: Estudiante
	Contraseña: *****
Resultado Esperado	El sistema debe permitir el ingreso e interacción, resolución o descarga de la tarea a realizar.
Condiciones	El usuario debe estar registrado dentro del curso para poder interactuar
Observación	La tarea permite la subida de un solo archivo, la respuesta con texto o la descarga de un archivo.
Procedimiento	Ingresar a la plataforma virtual
	Ingresar al curso donde se ha creado una tarea Ej.(Gestión del Desarrollo del Proceso de Software)
	Hacer click sobre el nombre de la tarea Ej.(ANÁLISIS DE PROBLEMAS")
	Hacer click sobre "Seleccionar archivo"

	Hacer click sobre "Subir este archivo"
	Hacer click sobre "Continuar"
Resultado Real	El sistema abre la página de la tarea que permite la interacción (subida de archivo) y regresa a la misma página luego de la carga.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de creación de cuestionarios calificados (Profesor)**

En esta prueba se realiza la creación de un cuestionario (examen, evaluación) por parte de un profesor de curso dentro de un modulo determinado del mismo.

Tabla 4.35. Prueba de Creación de Cuestionarios Calificados (Profesor)

Caso de Prueba	Creación de cuestionarios calificados (Profesor)
Entrada	Ingreso de Datos
	Nombre de Usuario: Profesor
	Contraseña: *****
Resultado Esperado	El sistema debe mostrar el modulo con el foro creado.
Condiciones	El usuario debe estar registrado como profesor del curso
Observación	El foro estará disponible desde su fecha de creación, de manera indefinida o por el periodo de días que el profesor lo considere.
Procedimiento	Ingresar a la plataforma virtual (Profesor)
	Ingresar al curso donde se desee crear el foro Ej.(Gestión del Desarrollo del Proceso de Software)
	En la sección de "Administración", hacer click sobre "Activar edición"

En el módulo que se desee crear la sala seleccionar dentro de la lista de "Agregar actividad..." la opción de

	"Cuestionario"
	Ingresar datos como "Nombre", "Introducción" y los ajustes adicionales como duración del examen, etc.
	Hacer click en "Guardar cambios y regresar al curso" o en "Guardar cambios y mostrar"
Resultado Real	Click sobre "Guardar cambios y regresar al curso": El sistema regresa a la pagina principal del curso con el ítem del examen creado dentro del modulo seleccionado.
	Click sobre "Guardar cambios y mostrar": El sistema redirecciona a la página del examen listo para resolverlo.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de rendición de cuestionarios calificados (Estudiante)**

En esta prueba se efectúa la rendición de un cuestionario por parte de un estudiante previamente registrado dentro del curso.

Tabla 4.36. Prueba de Rendición de Cuestionarios Calificados
(Estudiante)

Caso de Prueba	Rendición de cuestionarios calificados (Estudiante)
Entrada	Ingreso de Datos
	Nombre de Usuario: Estudiante
	Contraseña: *****
Resultado Esperado	El sistema debe permitir rendir satisfactoriamente el examen.
Condiciones	El usuario debe estar registrado dentro del curso para poder interactuar dentro de él.
	El usuario puede ser profesor del curso
Observación	El examen consta de límite de tiempo.

Procedimiento	Ingresar a la plataforma virtual
	Ingresar al curso donde se ha creado un cuestionario o examen en línea Ej.(Gestión del Desarrollo del Proceso de Software)
	Hacer click sobre el nombre del cuestionario Ej.("EVALUACION MIS-001")
	Hacer click sobre "Comenzar"
	Resolver el cuestionario
	Hacer click sobre "Enviar todo y terminar"
	Hacer click sobre "Cerrar esta ventana"
	Hacer click sobre "Continuar"
Resultado Real	El sistema abre la página (pop-up) de la evaluación que permite la interacción (resolución), muestra la calificación obtenida y cierra la página luego de la terminación.

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

- **Prueba de consulta de calificaciones (Estudiante)**

En esta prueba el estudiante accede a su boleta de calificaciones, pudiendo consultar todas sus notas o solamente sus promedios de un curso en el que se ha inscrito previamente.

Tabla 4.37. Prueba de Consulta de Calificaciones (Estudiante)

Caso de Prueba	Consulta de notas (Estudiante)
Entrada	Ingreso de Datos
	Nombre de Usuario: Estudiante
	Contraseña: *****
Resultado Esperado	El sistema debe permitir el ingreso e interacción dentro del curso.
Condiciones	

El usuario debe estar registrado dentro del curso para

	poder interactuar dentro de él
	El usuario puede ser profesor del curso
Observación	El sistema permite la consulta de promedios y notas individuales.
Procedimiento	Ingresar a la plataforma virtual
	Ingresar al curso donde se desee consultar las calificaciones Ej.(Gestión del Desarrollo del Proceso de Software)
	Hacer click sobre "Calificaciones"
Resultado Real	El sistema abre la página del registro de calificaciones del estudiante

Fuente: PROAÑO, Marco

Elaborado por: PROAÑO, Marco

CAPITULO 5.

5. CONCLUSIONES Y RECOMENDACIONES.

Una vez finalizado este proyecto en el cuál he trabajado arduamente junto a la dirección de mi Director de Tesis y al Departamento de las TIC's de la escuela, puedo sentirme satisfecho del resultado de este proyecto que servirá de soporte para la Escuela Politécnica del Ejército Sede Latacunga que busca el fortalecimiento de los métodos de enseñanza-aprendizaje.

Para concluir con el presente trabajo, he considerado importante emitir algunos comentarios, conclusiones, y recomendaciones basadas en esta experiencia, que esperamos sean de utilidad para futuras generaciones, las mismas se detallan a continuación.

5.1.- CONCLUSIONES

- La adopción de una plataforma de software libre tiene ventajas significativas. Al igual que en los desarrollos propios, la institución queda liberada de la dependencia cerrada respecto al vendedor, puesto que el software se distribuye con código abierto y con licencia de modificación libre. Las instituciones pueden aprovechar su capacidad técnica-informática con el objeto de integrar o generar modificaciones sobre el código original de un LMS a fin de adaptarlo a sus gustos y necesidades. Además esta opción significa un ahorro de recursos económicos debido a que los productos de software libre suelen ser distribuidos gratuitamente.

- El éxito que un LMS pueda tener dependerá directamente del uso que le den los usuarios (profesores y estudiantes) a la plataforma y estar concientes que la educación con el apoyo de nuevas herramientas tecnológicas brindarán una educación más pertinente y de mejor calidad.
- La adopción de un LMS en la institución no solamente puede servir de apoyo a las modalidades a distancia, sino también a las modalidades presencial y semi-presencial como se demostrará con el paso del tiempo y el correcto uso del sistema, debido a las herramientas de comunicación que un LMS puede ofrecer.
- El desarrollo de Moodle está siendo conducido por una comunidad de usuarios cada vez más amplia y abierta a la participación, lo que ha dado lugar a una evolución del producto más rápida de lo previsto y al desarrollo de módulos y características adicionales en un período muy breve de tiempo.
- Como ciclo de desarrollo del proyecto se optó por utilizar las metodologías de desarrollo Extreme Programming (XP) conjuntamente y trabajando a la par con SCRUM, debido a que son procesos de desarrollo ligero centrados en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.
- Las herramientas utilizadas para el desarrollo del presente proyecto han sido en su mayoría Open Source, economizando de esta manera los costos de construcción y mantenimiento del sistema.

5.2.- RECOMENDACIONES

- Escoger y mantener a Moodle como entorno virtual de enseñanza-aprendizaje de la Escuela Politécnica del Ejército Sede Latacunga por su combinación de flexibilidad y sofisticación didáctica, su flexibilidad tecnológica, el dinamismo de su comunidad de desarrollo y por su facilidad de uso para estudiantes y profesores, sin dejar de lado el nivel de adaptabilidad con la plataforma Web utilizada actualmente por la institución.
- Se recomienda estar pendiente de los nuevos releases o versiones liberadas por Moodle, con el objeto de incrementar la funcionalidad general de la plataforma, así como también el continuo mantenimiento de éste sistema, pues los requerimientos de la institución van cada día en aumento.
- Se recomienda el uso de software libre, pues la tendencia actual es la utilización de estas herramientas que a parte de economizar totalmente la construcción de un sistema, facilitan en gran manera la misma, a más de que permiten reutilizar el código y optimizarlo, dando con esto un buen mantenimiento al sistema que se haya construido.

GLOSARIO DE TÉRMINOS

BLENDED LEARNING: proceso docente semipresencial

SCROM: (Sharable Content Object Reference Model). Conjunto de especificaciones para desarrollo, empaquetamiento y distribución de material educativo en cualquier momento y en cualquier lugar.

ISO/IEC 9126: Estándar internacional para la evaluación del Software.

E-LEARNING: Es el suministro de programas educacionales y sistemas de aprendizaje a través de medios electrónicos.

LMS: Learning Management System o lo que es lo mismo Sistema de gestión de aprendizaje. Es un software que automatiza la administración de acciones de formación. Un LMS registra usuarios, organiza los diferentes cursos en un catálogo, almacena datos sobre los usuarios, también provee informes para la gestión. Suministra al instructor un mecanismo para crear y distribuir contenido, monitorear la participación de los estudiantes y evaluar su desempeño. También suelen ofrecer a los estudiantes el uso de mecanismos de interacción como foros de discusión, videoconferencias o servicios de mensajería instantánea.

AICC: Aircraft Industry Computer-Based Instruction Consortium Es Un órgano responsable de mantener los estándares de programación computarizada. AICC también se refiere al formato de cumplimiento de estas normas que se utiliza a menudo en los cursos en línea y con el apoyo de muchos productos de e-learning.

PLUGIN: Programa que puede anexarse a otro para aumentar sus funcionalidades (generalmente sin afectar otras funciones ni afectar la aplicación principal). No se trata de un parche ni de una actualización, es

un módulo aparte que se incluye opcionalmente en una aplicación.

WIKIS: Es un concepto que se utiliza en el ámbito de Internet para nombrar a las páginas Web cuyos contenidos pueden ser editados por múltiples usuarios a través de cualquier navegador. Dichas páginas, por lo tanto, se desarrollan a partir de la colaboración de los internautas, quienes pueden agregar, modificar o eliminar información.

RELEASES: es el acto de hacer público un programa de software, un service pack, un hot fix, etc., u otro tipo de información (documentación por ejemplo) que se pone a la disposición del público

SOPORTE **LDAP:** Lightweight Directory Access Protocol (en español Protocolo Ligero de Acceso a Directorios) que hacen referencia a un protocolo a nivel de aplicación el cual permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

BETA-TESTERS: Un betatester es una persona encargada de evaluar un software antes de que salga definitivamente al mercado. Ese software se dice que está en estado de beta.

OPEN SOURCE: El software Open Source se define por la licencia que lo acompaña, que garantiza a cualquier persona el derecho de usar, modificar y redistribuir el código libremente.

IMS: (o Sistema de Gestión Instruccional) es un estándar con el objetivo de definir especificaciones técnicas para la interoperabilidad de las aplicaciones y servicios en el aprendizaje distribuido, cuyo nombre formal es IMS Global Learning Consortium.

RUP: El Proceso Unificado de Rational (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

MSF: Microsoft Solutions Framework, un conjunto de principios, modelos, disciplinas, conceptos y directrices para la entrega de soluciones de tecnología de la información de Microsoft.

CAPTCHA: Es el acrónimo de Completely Automated Public Turing test to tell Computers and Humans Apart (Prueba de Turing pública y automática para diferenciar máquinas y humanos). Prueba desafío-respuesta utilizada en computación para determinar cuándo el usuario es o no humano.

POP-UP: Ventanas emergentes en sitios Web. El término denomina a las ventanas que emergen automáticamente.

REFERENCIAS BIBLIOGRAFICAS

Libros

- RAJASINGAHAN, Lalita y TIFFIN, John, En Busca De La Clase Virtual: La Educación En La Sociedad De La Información, Primera Edición, Editorial Paidós Ibérica, Barcelona, 1997, Pág. 280.
- AGUADED, Gómez, J.I. y Cabero Almenara, J. (dirs.). Educar en red: Internet como recurso para la educación. Málaga: Ediciones Aljibe, 2002.
- SCHWABER, Ken y BEEDLE, Mike, Agile Software Development With Scrum, Primera Edición, Editorial Prentice Hall, Estados Unidos, 2007, Pág. 158.
- MARTIN, Robert Cecil, Agile Software Development, Primera Edición, Editorial Prentice Hall, Estados Unidos, 2002, Pág. 529.
- PICHLER, Roman, Agile Product Management With Scrum, Primera Edición, Editorial Addison-wesley, Estados Unidos, 2010, Pág. 168.
- CHROMATIC y APANDI, Diaz Tatiana (edt), Extreme Programming, Primera Edición, Editorial Oreilly & Associates Inc, Estados Unidos, 2003, Pág. 90.
- BECK, Kent; ANDRES, Cynthia ; GAMMA, Erich (frw), Extreme Programming Explained, Segunda Edición, Editorial Addison-wesley, Estados Unidos, Pág 189.
- BECK, Kent; GAMMA, Erich (frw), Una explicación de la Programación Extrema: Aceptar el Cambio, Primera Edición, Pearson Educación, Madrid, 2002, Pág. 216.
- NEWKIRK, Martin, Programacion Extrema En La Practica, Primera Edición, Editorial Pearson Educación, Madrid, 2002, Pág.200.
- TRAMULLAS, Saz Jesús; GARRIDO Picazo Piedad, Software Libre para Servicios de Información Digital, Primera Edición, Editorial Pearson Educación, Madrid, 2006, Pág.400.

- YEFIM, Kats, Learning Management System Technologies And Software Solutions For Online Teaching, Primera Edición, Editorial Igi Global, Estados Unidos, 2010, Pág. 486.
- BARBERA, Elena; BADIA, Antoni, Educar Con Aulas Virtuales, Primera Edición, Editorial Machado Libros, España, 2004. Pág. 200.

Proyectos de Titulación

- MOSQUERA, Hilda, SEGOVIA Luis, El aula virtual: Teoría y aplicaciones. 2000
- LLUGSHA, Javier, Desarrollo De Nuevas Funcionalidades En Un Sistema Open Source De Aulas Virtuales (LMS), 2007
- CARRILLO, Germán, Análisis y Diseño de un sistema de aulas virtuales para el programa académico de pregrado y postgrado en administración pública para el fortalecimiento institucional de gobiernos seccionales autónomos, 2007.

Web

- http://www.guidance-research.org/sigossee/join/sp/docs/metodologia/evaluacion_pdf/attach/Quantified_methodology.pdf
- <http://www.dokeos.com/>
- www.ossite.org
- <http://www.moodle.org>
- <http://www.atutor.ca>
- www.opensourcecms.com
- www.joomlaspanish.org
- <http://www.gnu.org/copyleft/gpl.html>
- <http://www.manifiestoagile.com>

- www.xprogramming.com
- www.info-ab.uclm.es
- <http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf>
- <http://faculty.salisbury.edu/~xswang/Research/Papers/SERelated/scrum/ScrumXP.pdf>
- <http://www.dosideas.com/wiki/Categor%C3%ADa:Scrum>
- <http://www.orbisoft.com/>