

ESCUELA POLITÉCNICA DEL EJÉRCITO

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**DESARROLLO DE UN CURSO E-LEARNING
UTILIZANDO EL MÉTODO ERGOGLÍFICO
APOYADO EN UN CD INTERACTIVO
CASO PRÁCTICO: ASIGNATURA DE INGENIERÍA DE
SOFTWARE I**

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

**POR:
GABRIELA ALEXANDRA DEL CASTILLO AGUIRRE
HÉCTOR ALEJANDRO CARRERA QUERA**

SANGOLQUÍ, 27 FEBRERO DE 2009

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por los Sres. Héctor Alejandro Carrera Quera y Gabriela Alexandra Del Castillo Aguirre como requerimiento parcial a la obtención del título de **INGENIERO EN SISTEMAS E INFORMÁTICA**, bajo nuestra supervisión.

Sangolquí, 26 de Febrero del 2009

Ing. Edgar Hermosa.
DIRECTOR DE TESIS

Ing. Rodrigo Fonseca.
CODIRECTOR DE TESIS

INDICE

CAPITULO I	1
ASPECTOS GENERALES	1
1 1 Introducción	1
1 2 Justificación	2
1 3 Objetivos	3
1 3 1 Objetivo General	3
1 3 2 Objetivos Específicos	3
1 4 Alcance	4
1 5 Metodologías	5
1 6 Herramientas	5
1 7 MARCO TEÓRICO	7
1 7 1 Breve Historia	7
1 7 2 Definición de e-Learning	8
1 7 3 Mitos del e-learning	10
1 7 4 Características del e-learning	11
1 7 5 e-Learning vs Profesores	11
1 7 6 AULA VIRTUAL	12
1 7 6 1 VENTAJAS	13
1 7 7 Implantar un aula virtual	13
1 7 8 CAMPUS VIRTUAL	14
1 7 8 1 Características de Campus Virtual	15
1 7 9 METODOLOGÍAS	15
1 7 9 1 MODELO OOHDM	15
1 7 9 2 MÉTODO ERGOGLÍFICO	16
1 7 9 2 1 Fases del método Ergoglífico	17
1 7 10 DISEÑO DE CONTENIDOS	19
1 7 11 MAPAS DE NAVEGACIÓN	19
1 7 12 GUIONES MULTIMEDIA	20
1 7 12 1 Principios para la elaboración de un guión multimedia educativo	21
1 7 13 SOFTWARE EDUCATIVO	22
1 7 13 1 Estándares para Sistemas Educativos	23
CAPÍTULO II	25
2 1 Desarrollo del Material del curso de e-learning	25
2 2 División de la materia en lecciones	26
2 3 Desarrollo del Contenido de cada Lección	30
2 3 1 LECCIÓN No 1: Conceptos, evolución y mitos de la Ingeniería de Software	30
2 3 2 LECCIÓN No 2: Modelos del Proceso de Software	40
2 3 3 LECCIÓN No 3: Fundamentos de Gestión de Proyectos	58
2 3 4 LECCIÓN No 4: Método de estimación de puntos de función	71
2 3 5 LECCIÓN No 5: Relación entre los puntos de función y COCOMO	89
2 3 6 LECCIÓN No 6: Planificación de Proyectos	120
2 3 7 LECCIÓN No 7: Administración de Riesgos	128
CAPITULO III	138
3.1. ETAPAS	138
3.1.1. Diseño Conceptual	138
3.1.2. Mapas de Navegación	139
3.1.3. Diseño de interfases abstractas	140

3.1.4. Implementación	140
3.1.5. Casos de uso	141
3.2. ETAPA DE ANÁLISIS	151
3.2.1. Conocimiento de los usuarios	151
3.2.2. Identificación de las necesidades de conocimiento	152
3.2.3. Establecimiento de objetivos	153
3.2.4. Elaboración del diagrama de análisis	157
3.3. ETAPA DE DISEÑO	158
3.3.1. SELECCIONAR LA ESTRATEGIA	160
3.3.2. Estrategias Populares de Conocimiento	160
3.3.3. SELECCIONAR EL TIPO DE PRODUCTO DE CONOCIMIENTO	163
3.3.4. Asíncrono vs Sincrono	163
3.3.5. Tamaño de la clase	164
3.3.6. Escoger el tipo de Curso	164
3.3.7. ESPECIFICACIÓN DE LOS OBJETOS DE CONOCIMIENTOS	164
3.3.8. Definición del Objeto de conocimiento	164
3.3.9. Contenido de un objeto de Conocimiento	164
3.3.10. Cuerpo de una Página (Objeto de Conocimiento de Nivel Inferior)	165
3.3.10.1. Cuerpo de las lecciones (Objeto de conocimiento de nivel medio)	166
3.3.10.2. Jerarquía de objetos de conocimiento	167
3.3.10.3. Diagrama del Contenido del Curso	169
3.3.10.4. Objeto de Conocimiento de Nivel Superior	170
3.3.10.5. Objeto de Conocimiento de nivel medio	170
3.3.10.6. Objeto de Conocimiento de nivel Inferior	171
3.3.11. DISEÑO DETALLADO DEL CONTENIDO	184
3.3.11.1. Especificación del Módulo de Contenido	185
3.3.11.2. Especificación de Test	186
3.3.11.3. Especificación de las Actividades de Aprendizaje	188
3.3.12. ESPECIFICACIÓN DEL FRAMEWORK	191
3.3.12.1. Definición del frameWork	191
3.3.12.2. Objetivo del FrameWork	191
3.3.13. LISTADO DE LOS REQUERIMIENTOS TECNOLÓGICOS	192
3.3.13.1. Participantes y que necesitan ellos	194
3.3.13.2. Herramientas y tareas para cada Nivel	195
3.3.13.3. Herramientas para el curso	195
3.3.13.4. Especificación de formatos archivos	196
3.4. ETAPA DE CONSTRUCCIÓN	196
3.4.1. DISEÑO DEL FLUJO DE TRABAJO	198
3.4.1.1. Flujo de Trabajo de Producción	198
3.4.1.2. Utilización de plantillas	199
3.4.1.3. Utilización de Scripts	199
3.4.2. ESTABLECIMIENTO DE ESTÁNDARES INTERNOS DE INTERFACES	200
3.4.2.1. Selección de Colores	201
3.4.2.2. Selección del Tipo de Letra	201
3.4.2.3. Disposición de Frame	202
3.4.3. Utilización de Plantillas	203
3.4.3.1. Proceso para la creación de una plantilla	205
3.4.3.2. Plantillas necesarias	205
3.4.3.3. Plantillas para las páginas de Evaluación	206
3.4.3.4. Ensamblaje del Curso	214

3.5.	ETAPA DE EVALUACIÓN	215
3.5.1.	Visión general del proceso de evaluación	215
	CAPITULO IV	217
4.1.	Levantamiento del curso en la Plataforma e-learning	217
4.2.	Creación del curso	217
4.3.	Levantamiento del inicio del curso	220
4.4.	Levantamiento del de un documento	220
4.5.	Creación de cuestionarios	223
4.6.	Foro	228
4.7.	Glosario	231
	CAPITULO V	235
5 1	Conclusiones	235
5 1	Recomendaciones	236

INDÍCE DE FIGURAS

Figura 1.1:	Áreas del e-learning	8
Figura 1.2:	Comparando los papeles de un Profesor de Aula de Clases y de un Equipo de trabajo de e-learning	11
Figura 1.3:	Fases del Método Ergoglífico	16
Figura 1.4:	Repetición y Perfeccionamiento	17
Figura 1.5:	Estructura en red	18
Figura 1.6:	Esquema de la narración audiovisual interactiva	20
Figura 1.7:	Teorías de Aprendizaje	22
Figura 2.1:	Competencias del curso e-learning	25
Figura 3.1.	Etapas de la metodología OOHDM	138
Figura 3.2	mapa conceptual de un curso	138
Figura 3.3.	Mapa navegacional del Curso	140
Figura 3.4	Casos de uso Foro	142

Figura 3.5 Casos de uso Chat	143
Figura 3.6 Casos de uso Recursos	144
Figura 3.7. Fase de Análisis	152
Figura 3.8 Establecimiento del Objetivo General	153
Figura 3.9 Establecimiento de Objetivos Específicos	154
Figura 3.10 Diagrama de Diseño	159
Figura.3.11 Lecciones	166
Figura 3.12 Jerarquía de Objetos de Conocimiento	168
Figura 3.13 Framework Moodle	192
Figura 3.14. Participantes y Procesos	193
Figura 3.15. Necesidades de los Participantes	194
Figura 3.16 Herramientas y Tareas para Cada Nivel	195
Figura 3.17 Construcción	197
Figura 3.18. Flujo de Trabajo de Producción	198
Figura 3.19. Plantilla de trabajo	199
Figura 3.20. Script de programación	200
Figura 3.21 Plantilla para una página	204
Figura 3.22 Uso de la Plantilla General	204
Figura 3.23 Base para las plantillas	205
Figura 2.24. Ensamblaje del Curso	214
Figura 2.25. Proceso de Evaluación	215
Figura 4.1 Login de usuario	217
Figura 4.2 Agregar cursos	217
Figura 4.3 Datos del cursos	218
Figura 4.4 Datos del cursos	218
Figura 4.5 Enlazar una archivo a una Web	220
Figura 4.6 Datos para enlazar un archivo	221
Figura 4.7 Datos para enlazar un archivo	221
Figura 4.8 Dirección del archivo	222
Figura 4.9 Dirección del archivo	223
Figura 4.10 Cuestionario	223
Figura 4.11 Cuestionario	224
Figura 4.12 Preguntas de un cuestionario	226
Figura 4.13 Categorías de una pregunta	226
Figura 4.14 Categorías de una pregunta	227
Figura 4.15 Cuestionario levantado en el curso	227
Figura 4.16 Foro	228
Figura 4.17 Foro en el curso	231
Figura 4.18 Glosario	231
Figura 4.19 Creación de un Glosario	232
Figura 4.20 Glosario de Términos	234

INDÍCE DE TABLAS

Tabla 1.1: Factores de implantación de un aula virtual	12
Tabla 1.2: Principios para elaborar un guión multimedia	20
Tabla 1.3: Estándares de Sistemas Educativos	23
Tabla 3.1 Especificación caso de uso – Ver Foro	145
Tabla 3.2 Especificación caso de uso – Crear categoría o tipo de foro	145
Tabla 3.3 Especificación caso de uso – Modificar categoría de foro	145
Tabla 3.4 Especificación caso de uso – Eliminar categoría de foro	145
Tabla 3.5 Especificación caso de uso – Agregar mensaje o tema en foro	146
Tabla 3.6 Especificación de caso de uso – Modificar mensaje o tema en foro	146
Tabla 3.7 Especificación caso de uso – Eliminar un mensaje o tema en el foro	146
Tabla 3.8 Especificación caso de uso – Crear Chat	146
Tabla 3.9 Especificaciones de caso de uso – eliminar Chat	147
Tabla 3.10 Especificaciones de caso de uso – Ver Chat	147
Tabla 3.11 Especificaciones de caso de uso – Insertar/ enviar mensaje en el Chat	147
Tabla 3.12 Especificación de caso de uso – Responder mensaje de Chat	147
Tabla 3.14 Especificaciones de caso de uso – Conectar Chat	148
Tabla 3.15 Especificaciones de caso de uso – desconectar Chat	148
Tabla 3.16 Especificación de casos de uso – Agregar recurso a lección o módulo	148
Tabla 3.17 Especificación de casos de uso – Ver recurso de módulo o lección	148
Tabla 3.18 Especificación de casos de uso – Modificar recurso de módulo o lección	149
Tabla 3.19 Especificación de casos de uso – Modificar recurso de módulo o lección	149
Tabla 3.20 Especificación de casos de uso – Eliminar recurso de módulo o lección	149
Tabla 3.21 Especificación de casos de uso – Descargar recurso de módulo o lección	149
Tabla 3.22. Verbos para el Planteamiento de Objetivos	154
Tabla 3.23 OBJETIVOS	157
Tabla 3.24 Diagrama de análisis	158
Tabla 3.25 fortalezas y debilidades	162

Tabla 3.26 Diagrama del contenido del curso	169
Tabla 3.27 Objeto de conocimiento de nivel superior	170
Tabla 3.28 Objeto de Conocimiento de nivel medio	171
Tabla 3.29 Objeto de Conocimiento de nivel Inferior	172
Tabla 3.30 Objeto de Conocimiento de nivel Inferior	173
Tabla 3.31 Objeto de Conocimiento de nivel Inferior	174
Tabla 3.32 Objeto de Conocimiento de nivel Inferior	175
Tabla 3.33 Objeto de Conocimiento de nivel Inferior	176
Tabla 3.34 Objeto de Conocimiento de nivel Inferior	177
Tabla 3.35 Objeto de Conocimiento de nivel Inferior	178
Tabla 3.36 Objeto de Conocimiento de nivel Inferior	179
Tabla 3.37 Objeto de Conocimiento de nivel Inferior	180
Tabla 3.38 Objeto de Conocimiento de nivel Inferior	181
Tabla 3.39 Objeto de Conocimiento de nivel Inferior	182
Tabla 3.40 Objeto de Conocimiento de nivel Inferior	183
Tabla 3.41 Objeto de Conocimiento de nivel Inferior	184
Tabla 3.42 elementos de media (animación)	185
Tabla 3.43 Especificación de Test	186
Tabla 3.44 Especificación de Test	187
Tabla 3.45 Actividad de Aprendizaje	188
Tabla 3.46 Actividad de Aprendizaje	189
Tabla 3.47 Actividad de Aprendizaje	190
Tabla 3.48 formato de archivo	196
Tabla 3.49 Paleta de colores	201
Tabla 3.50 Tipografía	202
Tabla 3.51 Frame	203
Tabla 3.52. Página Bienvenida	206
Tabla 3.53. Página Bienvenida	207
Tabla 3.54. Página de Resumen	208
Tabla 3.55. Página de Reflexión	209
Tabla 3.56. Página de Actividad de Aprendizaje	210
Tabla 3.57. Página de Actividad de Aprendizaje	211
Tabla 3.58. Página de Actividad de Aprendizaje	212
Tabla 3.59. Página de Actividad de Aprendizaje	213
Tabla 4.1 Descripción del foro	229

CAPITULO I

ASPECTOS GENERALES

1.1 Introducción

A diario se escucha que la gran mayoría de los problemas cotidianos de la sociedad, tienen relación directa con las deficiencias en la educación que se imparte y se recibe; por otro lado la alta deserción de los estudiantes de las aulas de aprendizaje, se debe precisamente a aspectos derivados de la motivación, a falta de recursos y a la poca innovación en los procesos educativos.

Los continuos cambios que se presentan en la sociedad actual producto de una globalización en todos los órdenes, hacen que la tecnología cumpla con un importante papel, pues fomenta procesos de aprendizaje en los cuales se utilizan elementos tales como la pedagogía, el diseño de aprendizaje y el análisis de factores sociales y culturales. De ahí que, si no se tiene muy clara esta visión de los sistemas de aprendizaje, no se aprovechará todo el potencial de la tecnología.

En la actualidad innovar en el área educativa es esencial y fundamental, por esta razón el e-learning o formación en línea es una propuesta de aprendizaje basada en la utilización de recursos pedagógicos desarrollados a partir de nuevas tecnologías.

Esta tendencia que en países denominados de primer nivel viene siendo utilizada de tiempo atrás, de ahí su desarrollo, poco a poco pretende ser utilizada en el Ecuador, planteando la utilización de una herramienta de autoformación que se adapte a los espacios, tiempos, ritmos y lugares de aprendizaje, de esta manera, se proporciona mayor flexibilidad tanto a los docentes como a los educandos.

El presente proyecto se enfoca en el análisis y utilización de esta metodología como un medio de enseñanza y pretende proporcionar a los participantes un intercambio dinámico que, conjuntamente con la voluntad y compromiso que exige este tipo de educación, permita obtener un aprendizaje de calidad.

Para lograr esa aspiración se propone el diseño y elaboración de un curso e-learning con el apoyo de un CD de los temas relacionados con la asignatura de Ingeniería de Software I impartida actualmente en la carrera de Ingeniería de Sistemas e Informática de la Escuela Politécnica del Ejército.

1.2 Justificación

El Departamento de Ciencias de la Computación de la Escuela Politécnica del Ejército, como parte de sus ramas de formación a los estudiantes, imparte la asignatura de Ingeniería de Software I en los niveles superiores de la carrera. Esta materia se dicta en la modalidad presencial debido a la complejidad de los contenidos y toda vez que no existe el material de apoyo suficiente que permita la formación en línea de los estudiantes, se hace necesario desarrollar este curso.

Espe virtual es el escenario en que se da los contenidos de esta materia es óptimo a nivel presencial, sin embargo si se desea utilizar una modalidad de e-learning dicho escenario debe contar con los medios tecnológicos necesarios que, a través de la innovación y mejoramiento de los métodos didácticos empleados, permitan obtener una educación integral y de calidad.

La elaboración de un curso e-learning y el CD multimedia permitirá obtener los conocimientos necesarios sobre la asignatura bajo una modalidad no presencial y además podrá ser utilizada para reafirmar los conocimientos adquiridos de manera presencial.

Para el desarrollo de este proyecto se plantea la utilización de distintas metodologías que apoyen el desarrollo de aplicaciones multimedia, como son: OOHDM, diseño de contenidos, mapas de navegación, guiones multimedia y se utilizará técnicas de ingeniería de software para sistemas educativos.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un curso e-learning apoyado con un CD multimedia que contenga los temas impartidos en la asignatura de Ingeniería de Software I, de la carrera de Ingeniería en Sistemas e Informática de la Escuela Politécnica del Ejército, utilizando herramientas actuales que permitan una óptima formación en línea de los educandos.

1.3.2 Objetivos Específicos

- Definir el contexto de la metodología OOHDm en que se desarrollará el curso e-learning para la asignatura de Ingeniería de Software I.
- Estudiar el método ergoglífico para el desarrollo de los contenidos del curso e-learning para la asignatura de Ingeniería de Software I.
- Especificar los contenidos que conformarán el presente tutorial, de manera que al estudiante logre obtener el conocimiento suficiente para desarrollarse en esta rama de la Ingeniería de Sistemas.
- Especificar la metodología del curso en línea apoyado en CD para la asignatura de Ingeniería de Software I, de manera que brinde el conocimiento suficiente para cumplir con los requerimientos de calidad educativa de la Escuela Politécnica del Ejército.
- Desarrollar un curso en línea mediante el adecuado manejo de las herramientas y metodologías a ser utilizadas.

1.4 Alcance

El proyecto se ejecutará siguiendo los métodos pedagógicos de e-learning y terminará con la elaboración de un CD multimedia. Los contenidos del curso

corresponden a la asignatura de Ingeniería de Software I dictada en la actualidad en la carrera de Ingeniería de Sistemas e Informática de la Escuela Politécnica del Ejército.

ESPE esta en proceso de reestructuración de las carreras, que actualmente se encuentra en una fase de rediseño del contenido de las materias, por el cual durante el desarrollo de esta tesis puede surgir cambios del contenido de la misma.

Los contenidos corresponden a los siguientes temas:

- *Introducción.* En este capítulo se explicará sobre la evolución de las metodologías que se utilizan para la elaboración de software, las definiciones su técnica, su diseño, etc. Lo cual nos permitirá entender su progreso en el desarrollo de software.
- *Fases para desarrollar Software.* Este capítulo comprende las fases que debe tener el desarrollo del software así como la planificación, el análisis del problema, diseño, administración, gestión de proyectos, métodos de procesos, gestión de riesgos, pruebas de software, métricas, estimaciones, etc.

1.5 Metodologías

Las metodologías que se utilizarán para el desarrollo de este proyecto son:

- *Modelo OOHDM (Object Oriented Hypermedia Design Methodology).* Modelo que se utiliza para la elaboración de aplicaciones hipermedia con Orientación a Objetos.
- *Método Ergoglífico.* Permitirá crear una estructura de plantillas para el desarrollo de contenidos del curso.
- *Diseño de Contenido.* Nos ayuda a realizar un esquema de los posibles contenidos que conformarán el CD multimedia.

- *Mapa de Navegación.* Este método permite realizar un esquema de cómo se navegará a través del CD multimedia.
- *Guiones Multimedia.* Metodología que permite especificar la secuencia de los contenidos dentro de la aplicación.
- *Ingeniería de Software para Aplicaciones Educativas.* Técnicas que deben ser tomadas en cuenta a la hora de realizar el desarrollo de un proyecto multimedia.

1.6 Herramientas

- *Macromedia Studio 8.* Con esta plataforma se pretende obtener excelentes características y rendimiento para el desarrollo de aplicaciones multimedia, entre las que se encuentran: Dreamweaver 8, Flash Professional 8, Fireworks 8 y Flash Paper 2.
- *Dreamweaver 8.* Utilizar esta herramienta permite realizar el diseño, desarrollo y mantenimiento de aplicaciones multimedia basados en normas.
- *Flash Professional 8.* La nueva versión de esta herramienta incluye características que permiten la creación de aplicaciones con un alto rendimiento en efectos gráficos y controles animados que mejoran la interacción con los usuarios.
- *Fireworks 8.* Con esta herramienta se logra optimizar imágenes y gráficos obteniendo excelente calidad con el mínimo tamaño de compresión.
- *Flash Paper 2.* Esta herramienta permite convertir fácilmente documentos, presentaciones, hojas de cálculo y otros tipos de archivo en documentos Macromedia Flash.

- *Neobook*. Es una herramienta que permite la creación de publicaciones multimedia interactivas a través de la organización y de las pantallas y la compilación en archivos .exe que permiten la distribución vía CD.
- *HotPotates*. Esta herramienta permite la estructuración de test para el proceso de Autoevaluación.

1.7 MARCO TEÓRICO

1.7.1 Breve Historia

En realidad la enseñanza a distancia aparece en el siglo XVIII dando sus primeros pasos como una nueva forma de enseñanza. En el año de 1983 con el uso de los computadores y métodos tradicionales ayudaron para que el e-learning se fuera fortaleciendo. Los cursos en CD-ROM como material de apoyo aparecerían en el año 1984-1993.

En 1986 se comenzaron a diseñar, los primeros módulos de lo que entonces se conocía como “Enseñanza asistida por ordenador”.

Uno de los aliados del e-learning fue el correo electrónico, ayudando a enviar tutorías online por este medio. Con el paso del tiempo y a partir del año 2000 los avances tecnológicos convirtieron al e-learning en lo que hoy conocemos.

El primer sistema de software desarrollado especialmente para estos fines, fue el PLATO, desarrollado por la Universidad de Illinois. De los últimos datos se conoce que aún se usa el sistema, actualizado según las últimas tecnologías.

A finales de 2003 comienzan a desarrollarse programas de formación para colectivos diversos como médicos, farmacéuticos, auditores o incluso agricultores. Hasta ese momento la oferta se centraba en cursos de Ofimática, Idiomas y Habilidades para profesionales de empresas. Varios pasos fueron surgiendo como cursar una titulación universitaria oficial, un master o un postgrado a través de la Web. Las universidades fueron desarrollando modelos en torno al open source motivados por desarrollar metodologías propias.

1.7.2 Definición de e-Learning

Existen varias definiciones, pero la que más se acerca a nuestro objetivo de proyecto de tesis es la siguiente: “El e-learning es capacitación no presencial que, a través de plataformas tecnológicas, posibilita y flexibiliza el acceso y el tiempo en el

proceso de enseñanza-aprendizaje, adecuándonos a las habilidades, necesidades y disponibilidades de cada discente, además garantiza ambientes de aprendizaje colaborativos mediante el uso de herramientas de comunicación sincrónica y asincrónica, potenciando el proceso de gestión basado en competencias”^(1.1)

Como nos podemos dar cuenta tenemos un gran compromiso con los alumnos de la materia de Ingeniería de Software I, debido que debemos proporcionarles competencias y destrezas en cada lección de la materia, considerando los principios del e-learning.

El e-learning engloba tres áreas fundamentales:

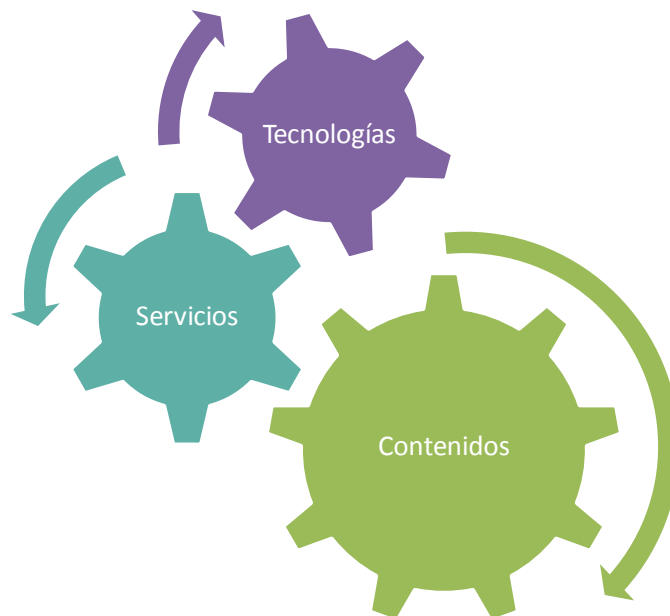


Figura 1.1: Áreas del e-learning

En primer lugar, en el ámbito de los Contenidos, la tendencia desarrollar pequeños Módulos o Unidades Didácticas reutilizables, que puedan combinarse entre sí en distintos planes formativos y en diferentes plataformas.

La utilización de multimedia en los contenidos es deseable, en tanto que parece demostrado que el aprendizaje se hace más efectivo si se ofrece la información

^(1.1) Gómez 2005

al estudiante en distintos formatos que se complementen entre sí: texto, imagen (gráfica o fotográfica), animaciones, sonido y vídeo.

Sin embargo, debido a las limitaciones en la velocidad de transmisión de contenidos por la Red, los formatos que ocupan gran cantidad de información como sonidos, ciertas imágenes y especialmente el vídeo, deben limitarse a los casos en que su uso esté plenamente justificado, en el sentido de que aporte algo específico al aprendizaje.

Por esto mismo, muchos proyectos de e-learning optan por desarrollar parte o la totalidad de los materiales en un soporte off-line como el CD-Rom, que ofrece una mayor velocidad de acceso a los contenidos.

En segundo lugar, y en relación a las Tecnologías, se debe considerar las plataformas y los estándares actuales. Entre las plataformas de e-learning están las herramientas que combinan hardware y software para ofrecer todas las prestaciones necesarias para la formación basada en la Red. Referente a los estándares, diferentes organizaciones trabajan para definirlos y asegurar la interoperabilidad de los distintos sistemas, y la reusabilidad de los contenidos en distintas plataformas.

En tercer lugar, el área de Servicios de apoyo a la implantación de un sistema de e-learning se considera la clave del éxito de este tipo de formación. El seguimiento y soporte ofrecidos al alumno durante el desarrollo del curso son aspectos fundamentales, y marcan la diferencia en la implantación y toma de acciones formativas.

1.7.3 Mitos del e-learning

Como en toda ciencia y nueva forma de enseñanza han surgido mitos que mal interpreta lo que el e-learning pretende:

MITO: Los cursos presenciales se convierten en e-learning al subirlos a la Web.

REALIDAD: el e-learning es una forma de trabajo distinta, e involucra metodologías para subir un curso y considerarlo e-learning.

MITO: Los cursos de e-learning son más baratos que la enseñanza presencial.
REALIDAD: el e-learning de calidad, implica un fuerte apoyo del recurso humano y no abarata la enseñanza.

MITO: el e-learning reemplazará a la formación presencial.

REALIDAD: la formación presencial y el e-learning son complementarios, pero no excluyentes, cada uno será un apoyo según las necesidades de los alumnos.

MITO: En un curso de e-learning no hay que ir a clase y la flexibilidad es total.

REALIDAD: En los cursos de e-learning existen objetivos y se establece las horas que el alumno dedicaría a cada unidad, también se debe ir a clase por medio del aula virtual. Lo que se pretende es tener una presencia diferida.

1.7.4 Características del e-learning

- Elimina las distancias físicas.
- Permite flexibilidad espacial y horaria.
- Favorece la interacción.
- Facilita el aprendizaje permanente durante toda la vida.
- El tutor atiende a varios alumnos en grupos reducidos.
- Creación compartida de conocimiento.
- Reutiliza los contenidos formativos.

1.7.5 e-Learning vs Profesores

E-learning no elimina a los profesores, pero si redefine sus roles o papeles. En el aula, es el profesor quien enseña, en e-learning, quien diseña el curso es el diseñador instruccional. ^(1.2)

Los métodos tradicionales de enseñanza de los instructores en el aula, se basa en leer, demostrar habilidades, contar historias, hacer preguntas, responder preguntas,

(1.2) Tomado del Libro: "(MarcadorDePosición1)E-Learning". William Horton Consulting.

asignar tareas, criticar trabajos, conducir pruebas, y dar notas. En cambio el equipo de trabajo de e-learning es seleccionar material, escribir textos, dibujar cuadros, mostrar videos, crear animaciones, programar interactividad, y configurar la tecnología. Ver Figura 1.2.

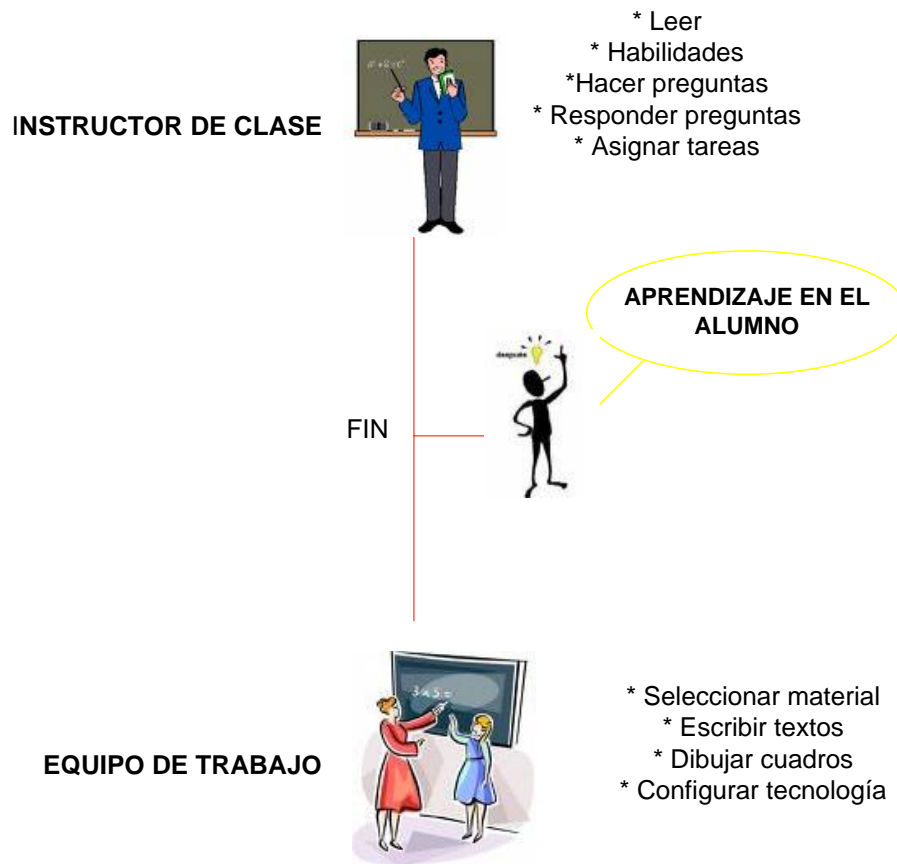


Figura 1.2: Comparando los papeles de un Profesor de Aula de Clases y de un Equipo de trabajo de e-learning. ^(1.3)

1.7.6 AULA VIRTUAL

Actualmente los sistemas de educación a distancia han dejado de ser sólo una alternativa más de enseñanza para convertirse en un modelo educativo de innovación pedagógica del presente siglo.

En la educación presencial el contacto es cara a cara con los alumnos, en espacios espacialmente diseñados llamados “aulas”, en la modalidad de educación a

^(1.3) Adaptado de ¹ Libro: E-Learning”. William Horton Consulting.

distancia, “el aula virtual” constituye el nuevo entorno de aprendizaje, convirtiéndose en un poderoso dispositivo de comunicación y de distribución de conocimientos que, además, ofrece un "espacio" para atender, orientar y evaluar a los participantes.

El aula virtual, está disponible las 24 horas del día por medio del Internet, ofrece los servicios y funcionalidades necesarias para el aprendizaje a distancia y responde a la necesidad de los docentes y alumnos de una comunicación directa y atención personalizada inmediata o diferida.

1.7.6.1 VENTAJAS

- Reduce notablemente los costos de la formación.
- No requiere de un espacio físico y evita los desplazamientos.
- Amplía notablemente su alcance dando mayores posibilidades a los que se encuentran más alejados de los Centros de formación.
- Permite el acceso a los cursos con total libertad de horarios.
- Proporciona un entorno de aprendizaje y trabajo cooperativos.
- Distribuye la información de forma rápida y precisa a todos los participantes.
- Se complementa, con la formación presencial.

1.7.7 Implantar un aula virtual

Los factores para implantar un aula virtual son las siguientes:

Tabla 1.1: Factores de implantación de un aula virtual

Factores	Preguntas que debe responder
Innovación	¿Existe ya una oferta formativa similar? ¿Qué ventajas competitivas ofrecerá este proyecto?
Rentabilidad	¿El grupo destinatario es lo suficientemente grande para cubrir, al menos, los costos de la inversión inicial?
Viabilidad	¿El grupo destinatario se adecuará realmente a una acción formativa a través de Internet?
Recursos	¿Existen suficientes recursos económicos y humanos para

El equipo de trabajo constituye un factor predominante dentro de cualquier proyecto y se recomienda que este constituido de la siguiente forma:

El Director del Proyecto; un Coordinador, Profesor; Tutor; Desarrollador de contenidos; proveedor de contenidos para el aprendizaje; Pedagogo; y el Desarrollador Técnico para la puesta en marcha e integración de contenidos en la web.

Cada curso que se desarrolle debe tener características particulares en función de las necesidades específicas a las que se intenta responder. Si bien hay Aulas Virtuales donde el nivel de integración y el intercambio entre los participantes en la web es máximo, y todos los recursos están disponibles en todo momento en internet, también hay otras en que podemos encontrar los materiales de formación en soporte papel.

Es relevante considerar que el éxito de la educación a distancia está sumamente relacionado con la elaboración del material didáctico y con la figura del tutor, quien acompaña y guía al estudiante para que no se desoriente. Los tutores capaces de animar a los estudiantes, responder de forma inmediata sus dudas, o aconsejarles en la búsqueda de información, haciendo posible que el alumno se sienta "acompañado" en su esfuerzo.

1.7.8 CAMPUS VIRTUAL

Permite aplicar el concepto de Educación Virtual partiendo de estrategias teleinformáticas basadas fundamentalmente en redes electrónicas de datos como INTERNET, facilitando además la implementación de esquemas para la Educación en Línea, y apoyo a la Educación a Distancia y Presencial.

El campus virtual se basa en los siguientes principios:

- Trabajo en equipo (colaborativo y cooperativo)
- Integración del conocimiento
- Ambientes de aprendizaje significativos
- Desarrollo integral (liderazgo, responsabilidad, autonomía)

1.7.8.1 Características de Campus Virtual

- Favorecen la administración de contenidos académicos por Facultades, carreras o unidades académicas en la diversidad de niveles y modalidades.
- Facilitan la racionalización de recursos informáticos
- Fomentan la descentralización de la información
- Fomenta la cultura informacional y el uso de TIC's entre los docentes y estudiantes
- Permite secuenciar las tareas de aprendizaje al ritmo individual de cada estudiante
- Facilita la retroalimentación.
- Atender un mayor número sujetos en formación.

1.7.9 METODOLOGÍAS

1.7.9.1 MODELO OOHDM

El modelo OOHDM (Object Oriented Hypermedia Design Methodology), para diseño de aplicaciones hipermedia y para la Web, fue diseñado por D. Schwabe, G. Rossi, and S. D. J. Barbosa. Es usada para diseñar diferentes tipos de aplicaciones hipermedia como presentaciones multimedia y, lo más importante numerosos sitios web.

(1.4)

OOHDM propone el desarrollo de aplicaciones hipermedia mediante un proceso de 4 etapas:

- diseño conceptual
- diseño navegacional
- diseño de interfaces abstractas
- implementación

El diseño conceptual construye un esquema representado por los objetos de dominio o clases y las relaciones entre dichos objetos. Un esquema conceptual basado en clases, relaciones y subsistemas es el que propone el modelo OOHDM.

^(1.4) Fuente: <http://es.wikipedia.org/wiki/OOHDM>

En la segunda etapa, el diseñador define clases navegacionales tales como nodos, enlaces y estructuras de acceso. Los enlaces se derivan de las relaciones y los nodos representan ventanas lógicas (views) sobre las clases conceptuales.

En el diseño de las interfaces abstractas se define la forma en la cual deben aparecer los contextos navegacionales. También se incluye el modo en que dichos objetos de interfaz activarán la navegación y el resto de funcionalidades de la aplicación, esto es, se describirán los objetos de interfaz y se los asociará con objetos de navegación.

La última etapa se refiere a la implementación y consiste en poner en práctica los objetos de interfaz con los objetos de implementación.

1.7.9.2 MÉTODO ERGOGLÍFICO

El Método Ergoglífico genera un proceso de desarrollo amplio y específico ^(1.5).

Las razones por las cuales el Método Ergoglífico es el más utilizado en el desarrollo de cualquier producto de conocimiento, son las siguientes:

- Sirve para todo producto de conocimiento, no sólo para impartir educación y entrenamiento.
- El Método Ergoglífico cubre todos los niveles de Diseño desde toda una currícula hasta íconos y botones individuales.
- Permite desarrollar plantillas de diseño consistentes que pueden ser reutilizadas dentro de cualquier otro curso.
- Es el más utilizado porque permite desarrollar prototipos rápidamente, antes de tener un producto terminado.
-

1.7.9.2.1 Fases del método Ergoglífico

(1.5) Tomado de la Tesis: “Desarrollo de un curso WBT a nivel de prototipo para la fundación UCHLAM”; Esteban A. Cuña- César Villacís

El Método Ergológico se compone de cuatro fases principales: Análisis, Diseño, Construcción y Evaluación. Ver figura (1.3).

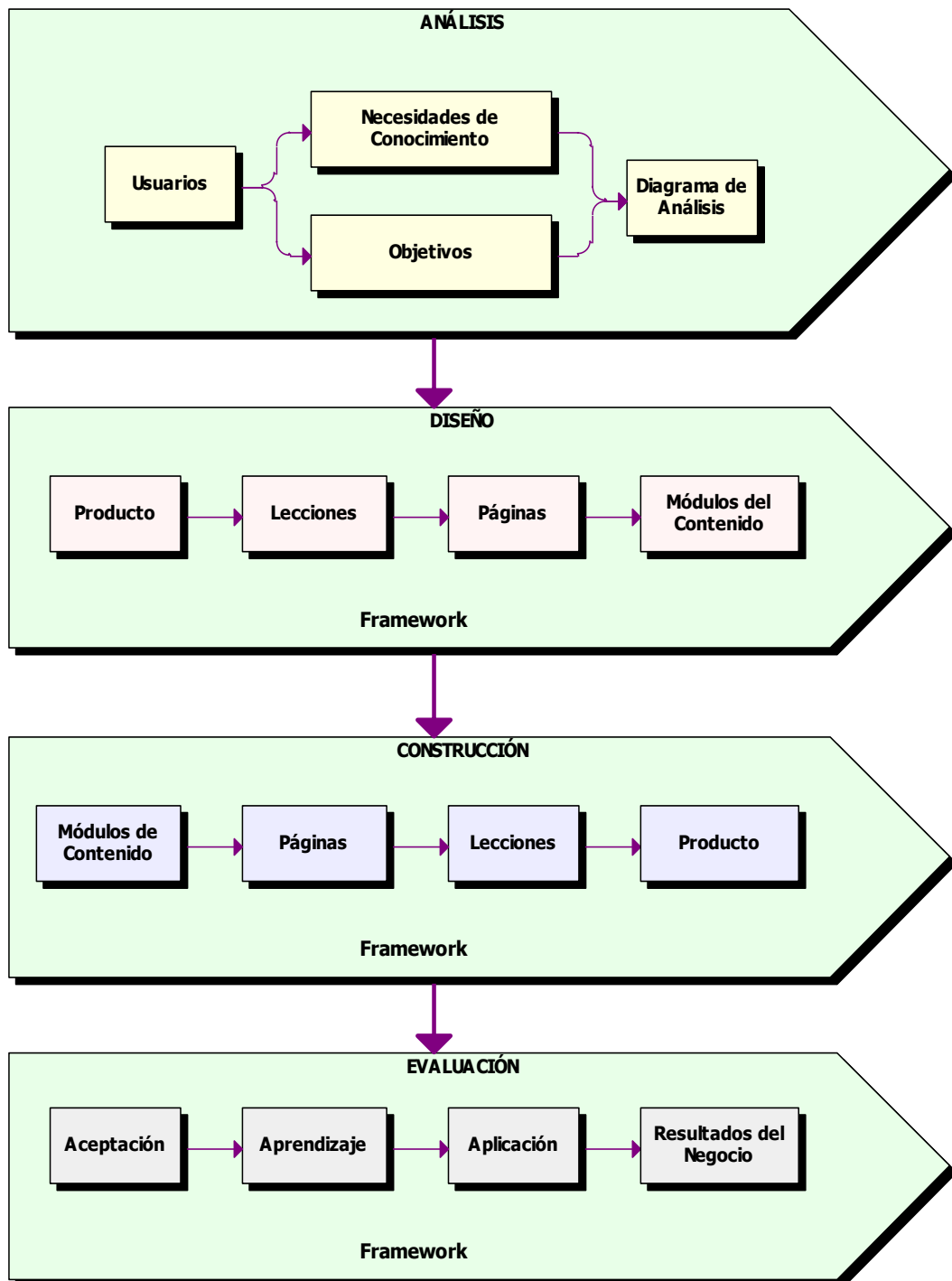


Figura 1.3: Fases del Método Ergológico. ^(1.6)

El análisis de la situación comienza con la definición del objetivo general del proyecto, de donde se derivan los objetivos específicos. El grupo de personas que va a utilizar este producto es determinado a partir de conocer los objetivos.

^(1.6) Adaptado del Libro: "Leading E-Learning". William Horton Consulting.

El diseño es esencialmente el refinamiento de los objetivos. La construcción se o puede realizar mientras se está diseñando. Creando primero pequeños componentes, los cuales se irán integrando hasta formar unidades grandes, para obtener finalmente el curso completo. Es necesario aclarar que el framework (perfil del curso) también debe hacerse paralelamente.

La fase de evaluación, se lo realizará cuando el curso esté completamente terminado, el mismo que tiene que ser probado por un grupo de personas quienes serán los encargados de evaluarlo en los niveles de aprendizaje alcanzado, aceptación, facilidad de uso y aplicación en el mundo real.

Todo este proceso de desarrollo es secuencial, se empieza por analizar la necesidad, se diseña y construye la solución para finalmente evaluarlo. El desarrollo de la mayoría de los productos de conocimiento es cíclico, lo que significa que se puede desarrollar a través de varios ciclos de análisis, diseño, construcción y evaluación, solo a través de estos ciclos es como se obtiene un curso e-learning de calidad.

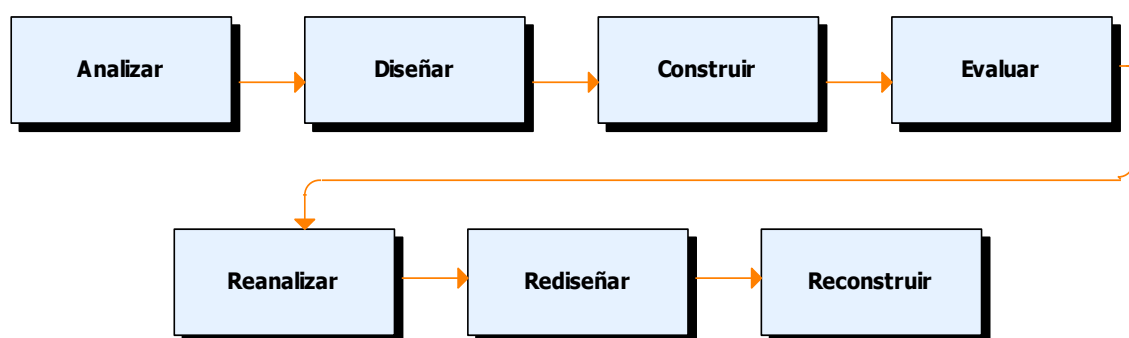


Figura 1.4: Repetición y Perfeccionamiento ^(1.7)

2.

1.7.10 DISEÑO DE CONTENIDOS

Los contenidos se pueden organizar de varias formas, entre las que tenemos:

- Estructura lineal

^(1.7) Adaptado del Libro: “Leading E-Learning”. William Horton Consulting.

Este tipo de diseño permite ir hacia delante o hacia atrás, por lo que se considerada una navegación limitada.

- Estructura jerárquica

Su estructura esta basada en una organización temática de contenidos, su navegación es hacia arriba o hacia abajo.

- Estructura en red

Esta navegación no es restringida, el alumno puede recorrer las páginas en cualquier sentido a través de múltiples enlaces. La cual resulta ser la más aconsejable. En vista que nuestro proyecto es desarrollar un curso de e-learning para la plataforma Moodle, la estructura en red se adapta a ésta configuración.

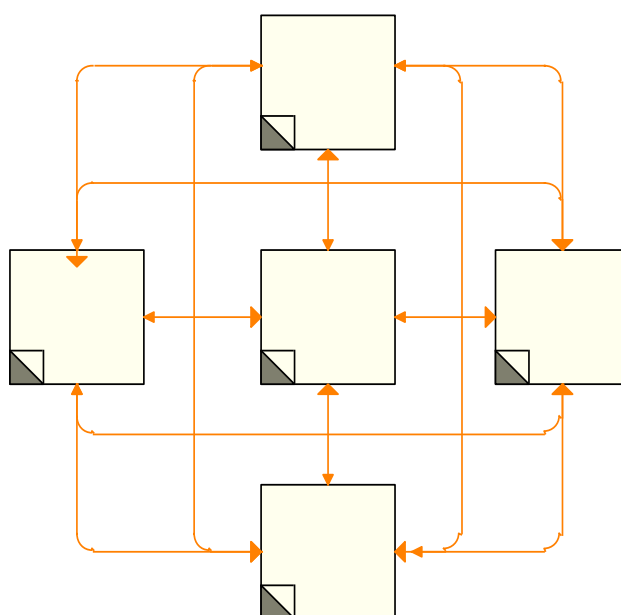


Figura 1.5: Estructura en red

1.7.11 MAPAS DE NAVEGACIÓN

En un mapa de navegación se modela la estructura de la información en un sitio Web permitiendo ver la relación que hay entre los documentos y archivos que forman parte del sitio y la jerarquía que hay entre ellos.

Define cada una de las pantallas, componentes conceptuales, distribución del espacio; así como también los nombres de los archivos reales y los adicionales, como formularios o multimedia y sus vínculos.

Un mapa de navegación se lo puede relacionar como un nodo con información sobre otros nodos, que incluye representaciones de la totalidad o parte de la información contenida en el hipertexto, con el fin de facilitar la navegación y el acceso a la información a los lectores del hipertexto.

Un mapa de navegación puede consistir en tres modelos básicos:

- Sumarios
- Índices
- Trayectos

Los sumarios son representaciones jerárquicas de la estructura del hipertexto, como la indicación de sus secciones principales y subsecciones. Los índices son listas de temas o propiedades relevantes del hipertexto, como los índices temáticos, de autores o de ilustraciones. Los trayectos son recorridos predefinidos recomendados por el autor del documento.

1.7.12 GUIONES MULTIMEDIA

El Guión Multimedia es un documento escrito que contiene una descripción detallada de todas y cada una de las escenas del producto audiovisual. Un guión, por lo tanto, no es otra cosa que una historia contada en imágenes. ^(1.8)

El guión es un elemento clave en el proceso de producción:

- Es jerárquicamente el elemento inicial de la producción.
- Es la herramienta directriz para la dirección técnica.
- Es una guía para los desarrolladores.
- Es una fuente para la previsión del presupuesto.

En el siguiente cuadro se observa el proceso en el que se encuentra el desarrollo del guión multimedia. (Ver Figura 1.6)

^(1.8) Fuente: <http://personal2.redestb.es/jevabe/guion.htm>

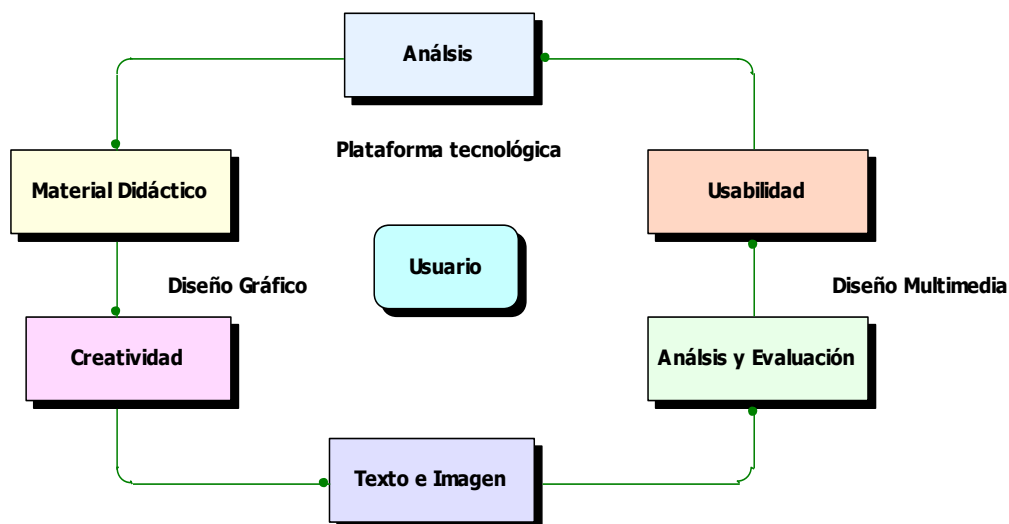


Figura 1.6: Esquema de la narración audiovisual interactiva^(1.9)

Un guión multimedia requiere de una exhaustiva investigación sobre el material que ha de utilizar. Es preciso documentarse racional, estética y científicamente para seleccionar, posteriormente, lo esencial. Un riguroso método de trabajo hace que el guionista no sea sólo un escritor sino un estudioso que descansa en el valor científico de los datos reales.

1.7.12.1 Principios para la elaboración de un guión multimedia educativo

Tabla 1.2: Principios para elaborar un guión multimedia

Principios	Descripción
Organización	En un guión multimedia el contenido o tema debe estar muy bien organizado para que la información sea fácilmente entendida.
Integración	Un buen multimedia logra una integración de la imagen, sonido y palabras escritas, tomando cada uno de ellos el protagonismo cuando es necesario. No debemos convertir el audiovisual en una conferencia ilustrada con imágenes, ni viceversa.
Narración	Introducir el tema a través de una historia, un contexto o la descripción de una situación cotidiana capta mejor la atención porque produce empatía y complicidad, genera sentimientos, mueve a la reflexión al provocar conflictos cognitivos o éticos e incrementa la curiosidad por conocer el desenlace. Un buen guionista debe ser un buen narrador, capaz de hacer creer al usuario del multimedia que lo que se le cuenta es real, creíble o verosímil.
Ritmo	El manejo del tiempo es un elemento esencial en el diseño y

(1.9) Fuente: http://www.nosolousabilidad.com/articulos/multimedia_elearning.htm

elaboración de un producto audiovisual. Aquí unos pocos segundos pueden convertirse en una eternidad. Debemos ser breves y concretos. Todos y cada uno de los elementos visuales, sonoros y textuales deben contener la información precisa y nada más.

1.7.13 SOFTWARE EDUCATIVO

La enseñanza asistida por computador se ha convertido en una rama de investigación importante de la inteligencia artificial. Francisco Ruedas menciona algunas técnicas, empleadas para desarrollar software educativo, tales como representación del conocimiento, sistemas expertos, redes neuronales y procesamiento de lenguaje natural.

Alvaro Galvís hace una referencia bastante completa de "Ingeniería de Software Educativo". En esencia determina que se debe conservar los pasos o etapas de un proceso sistemático para desarrollo de materiales (análisis, diseño, desarrollo, prueba y ajuste, implementación). Sin embargo, en este caso se da particular énfasis a los siguientes aspectos: la solidez del análisis, como punto de partida; el dominio de teorías sustantivas sobre el aprendizaje y comunicaciones humanas, como fundamento para el diseño de los ambientes educativos computarizados; la evaluación permanente y bajo criterios predefinidos, a lo largo de todas las etapas del proceso, como medio de perfeccionamiento continuo del material; la documentación adecuada y suficiente de lo que se realiza en cada etapa, como base para el mantenimiento que requerirá el material a lo largo de su vida útil.

Dentro de las teorías de aprendizaje existen dos tipos de tendencias marcadas:

- La primera está basada en el entorno del estudiante y se denomina conductivismo.
- La segunda se basa en el estudiante en sí y se la conoce como cognoscitivismo.

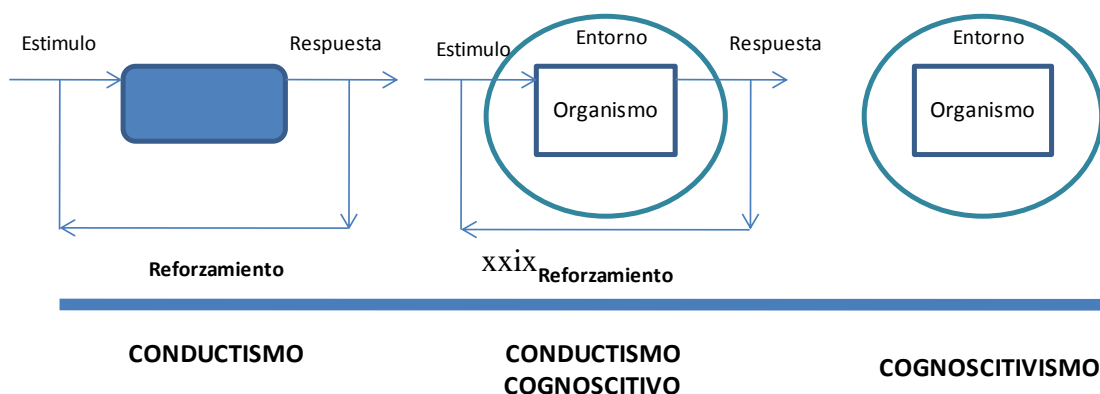


Figura 1.7: Teorías de Aprendizaje ^(1.10)

La unión de estas dos tendencias logra el equilibrio requerido para la asimilación de contenidos por parte de un participante en un curso de e-learning.

En el primer polo no se toma en cuenta al sujeto que aprende, sólo las condiciones externas que favorecen su aprendizaje; por esto se habla de un modelo de "caja negra" en el que lo fundamental es la programación en pequeños pasos, de eventos que conduzcan a lograr el resultado esperado (respuesta) y el reforzamiento de las respuestas, que confluyen hacia el logro de lo que se desea.

En el otro polo lo que cuenta es el individuo, el aprendiz, con todo su campo vital, su estructura cognoscitiva, las expectativas que tiene. Por contraposición se habla de un modelo de "caja traslúcida" en el que lo que cuenta es el aprendiz dentro de su entorno psicológico y social. La motivación interna, la significancia, el procesamiento de la información, las aptitudes de las personas, entre otros, son tomados en cuenta como factores que promueven el aprendizaje.

1.7.13.1 Estándares para Sistemas Educativos

Los estándares se ven necesarios en la educación en línea, dado el alcance global que tienen las aplicaciones e-learning por el uso de los sistemas de telecomunicación y al creciente interés de los individuos en la autoformación y en el aprendizaje a lo largo de toda la vida. Contar con aplicaciones estandarizadas marcará un mejor y mayor aprovechamiento de los esfuerzos de los cursos e-learning.

^(1.10) Fuente: <http://www.inf.udec.cl/revista/edicion6/psalcedo.htm>

Como se ha mencionado anteriormente, diferentes grupos de trabajo han desarrollado especificaciones y estándares orientados hacia una forma común de identificar, definir y comunicar a todos los recursos involucrados en un entorno e-learning (contenidos, docentes, estudiantes, aplicaciones, proveedores, etc. A continuación se describen brevemente algunos de ellos:

Tabla 1.3: Estándares de Sistemas Educativos

Estándares	Descripción
AICC (Aviation Industry Computer-Based Training Comitee) ^(1.11)	Es una asociación de entrenamiento profesional basado en tecnología, especializado en el sector de la aviación pero que se ha permeado también a otros sectores. Se reconoce como una de los precursores de la estandarización de materiales del entrenamiento profesional.
IMS (Global Consortium Inc) ^(1.12)	Investiga sobre las maneras de describir y codificar las metodologías de aprendizaje incorporadas en una solución e-learning. El estándar IMS-LD, identificado como un modelo conceptual de diseño instruccional, permite organizar un proceso de aprendizaje a partir de la definición de un escenario conformado de recursos educativos, servicios y actividades.
ADL (Advanced Distributed Learning) ^(1.13)	La misión de ADL es proveer acceso de la más alta calidad en educación y entrenamiento, en cualquier lugar y en cualquier momento. Para cumplir con estos objetivos.
ARIADNE	Es un proyecto de investigación y de desarrollo

^(1.11) Fuente: <http://www.aicc.org/>.

^(1.12) Fuente: <http://www.imsproject.org/>

^(1.13) Fuente: <http://www.adlnet.org>

<p>(Alliance of Remote Instructional Authoring and Distribution Networks for Europe) ^(1.14)</p>	<p>tecnológico de telemática para la educación y el entrenamiento, patrocinado por la Unión Europea. El proyecto se enfoca al desarrollo de herramientas y metodologías para producir, administrar y reutilizar elementos pedagógicos basados en computadora, así como la currícula de entrenamiento a distancia.</p>
<p>IEEE/LTSC (Institute of Electrical and Electronics Engineers/Learning Technology Standards Committee) ^(1.15)</p>	<p>El IEEE es una asociación internacional, cuya misión es promover los procesos ingenieriles para la creación, desarrollo, integración, compartición y aplicación del conocimiento sobre tecnologías electrónicas y de información. Dentro de su organización cuenta con el Comité de Estándares para Tecnología del Aprendizaje o LTSC, que se encarga de desarrollar estándares técnicos, recomendaciones y guías para la tecnología educativa.</p>
<p>W3C (World Wide Web Consortium) ^(1.16)</p>	<p>El consorcio W3C se encarga del desarrollo de tecnologías interoperables (especificaciones, normas, software y herramientas) para aprovechar todo el potencial del Web. Aunque este consorcio no está directamente vinculado con el desarrollo del e-learning es importante mencionarlo ya que de la interoperabilidad de la web dependen muchas de las funciones de la educación en línea.</p>

^(1.14) Fuente: <http://www.ariadne-eu.org>.

^(1.15) Fuente: <http://www.ieee.org>.

^(1.16) Fuente: <http://www.w3.org>.

CAPÍTULO II

2.1 Desarrollo del Material del curso de e-learning

El curso de e-learning a sido dividido en 7 lecciones, de tal manera que el contenido de la materia es distribuido desde lo esencial de la ingeniería de software con sus conceptos, por la mitad del curso viene la parte fuerte que son estimaciones y concluyendo con la gestión de riesgos.

Para cada lección se ha determinado sus competencias, las mismas que se muestran en la siguiente figura: 2.1

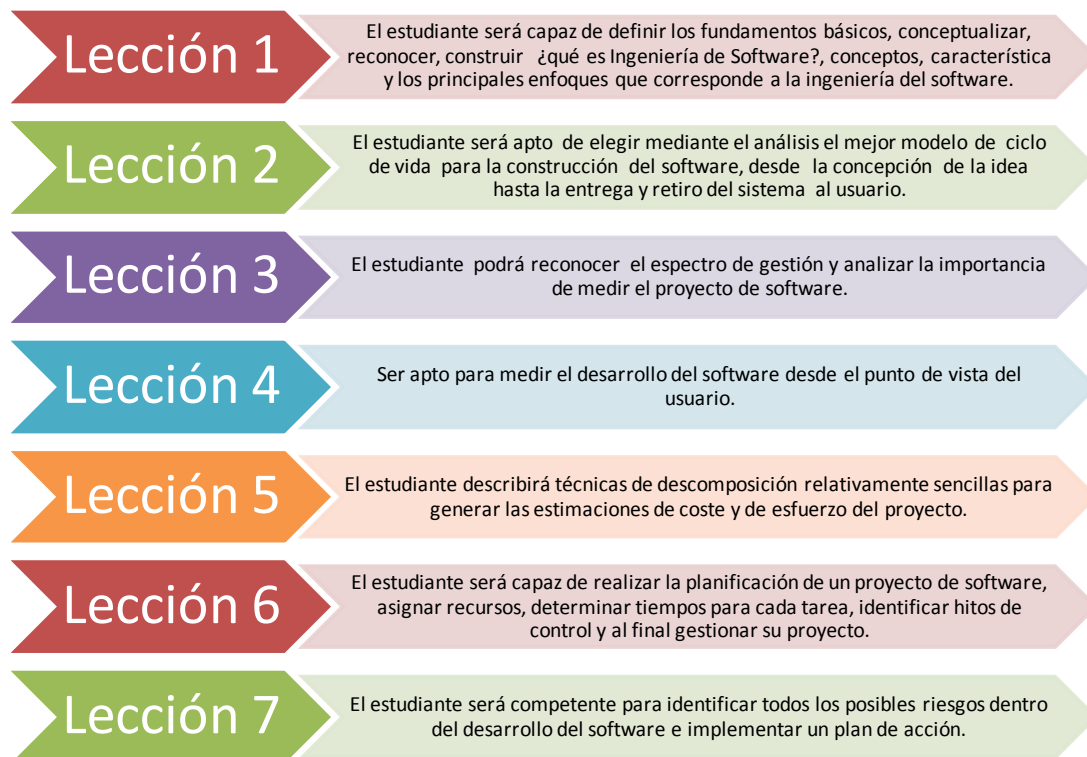


Figura 2.1: Competencias del curso e-learning

2.2 División de la materia en lecciones

Cada lección ha sido dividida según el material a ser recibido en forma presencial, tomando en cuenta que como es un curso e-learning debe ser lo más entendible posible para que pueda resolver las evaluaciones y los ejercicios sin

mucha ayuda del tutor. A continuación se muestra la distribución en lecciones del curso:

LECCIÓN No. 1

INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE

- Conceptos,
- Problemas del software
- Crisis del software
- Evolución del software
- Mitos del software
- Conceptos de ingeniería de Software
 - Objetivos
 - ¿Por que es importante ser ingeniero de software?
 - Importancia de la ingeniería del Software
 - Ejemplo de importancia de la ingeniería del Software
 - Características del software
 - Componentes de la ingeniería de software
 - Retos de la ingeniería de software
 - Aplicaciones del software

LECCIÓN No. 2

MODELOS DE PROCESO DE SOFTWARE

- Ciclo de Vida del Software
- Modelos de un Proceso de Software
 - Definición de un Modelo de Ciclo de Vida.
 - Etapas del Ciclo de Vida
 - Análisis
 - Diseño
 - Codificación
 - Pruebas
 - Mantenimiento
 - Modelo en Cascada
 - Modelo De Desarrollo Evolutivo
 - Modelo De Desarrollo Incremental
 - Modelo Espiral
 - Modelo DRA (Desarrollo Rápido de Aplicación)
 - ¿Cuál es el modelo más adecuado?
- Modelos de proceso de software
 - Modelos Tradicionales y modelos ágiles en el desarrollo de software
 - Metodologías pesadas estándar IEEE sobre proceso de software
 - Proceso de selección de un modelo de ciclo de vida del producto
 - Procesos de gestión del proyecto
 - Proceso de Iniciación del Proyecto
 - Proceso de seguimiento y control del Proyecto
 - Procesos orientados al desarrollo de software

- Procesos de Pre-Desarrollo
 - Proceso de Exploración de Conceptos
 - Proceso de Asignación del Sistema
- Procesos de Desarrollo
 - Proceso de Requisitos
 - Proceso de diseño
 - Proceso de implementación
- Procesos de Post-Desarrollo
 - Proceso de instalación
 - Proceso de operación y soporte
 - Proceso de mantenimiento
 - Proceso de retiro
- Procesos integrales del proyecto
 - Proceso de verificación y validación
 - Proceso de Gestión de la Configuración
 - Proceso de desarrollo de documentación
 - Proceso de formación
- Diferencia entre ciclo de vida y procesos de software

LECCIÓN No. 3

FUNDAMENTOS DE GESTIÓN DE PROYECTOS

- Gestión de Proyectos
 - Espectro de la Gestión
 - El personal
 - El producto
 - El Proceso
 - El Proyecto
 - Métricas del Software
 - Importancia de medir
 - ¿Qué son métricas de software?
 - Características
 - Beneficios
 - Métricas del Producto
 - Métricas orientadas al tamaño
 - Métricas de Proceso
 - Métricas de Seguimiento y control de proyectos
 - Métricas de Estimación
 - ¿Qué es una estimación?
 - Métricas de calidad

LECCIÓN No. 4

MÉTODO DE ESTIMACIÓN

- Método de estimación de puntos de función
 - Proceso del método de Puntos de Función
 - Obtención de los puntos de función sin ajustar
 - Parámetros Básicos
 - ILF Fichero Lógico Interno (Internal Logic File)

- Reglas de identificación:
- Ejemplos:
- EIF Fichero Lógico Externo (External Interface File)
 - Reglas de identificación:
- EI Entrada Externa (External Input)
 - Reglas de Identificación:
 - Ejemplo de entradas internas:
- EO Salidas Externas (External Output)
 - Reglas de Identificación
 - Ejemplo de salidas externas
- EQ Consultas Externas (External Query)
 - Reglas de Identificación:
 - Ejemplo de consultas externas:
- Valoración de la complejidad
 - Valoración de la complejidad de un ILF o EIF
 - Valoración de la complejidad de un EI o EO
 - Valoración de la complejidad de un EQ
- Definición del Factor de Ajuste
- Determinación del valor del grado de influencia GDI
- Obtención de los puntos de Función Ajustados

LECCIÓN No. 5

ESTIMACIÓN DEL PROYECTO DE SOFTWARE

- Relación entre los puntos de función y COCOMO
- COCOMO 81
- Modelo Básico
 - Cálculo del Esfuerzo
 - Cálculo del tiempo de desarrollo del proyecto
 - Cálculo de las personas necesarias para realizar el proyecto
 - Cálculo del costo total del Proyecto
 - Cálculo del esfuerzo de mantenimiento del software
- Modelo Intermedio
- Atributos
 - Atributos del producto
 - Atributos del ordenador
 - Atributos del personal
 - Atributos del proyecto
- Cálculo del Esfuerzo
- Modelo Detallado
- COCOMO II
 - Diseño Inicial y Post-Arquitectura
 - Estimación del Esfuerzo Nominal
 - Ajuste mediante drivers de Costo
 - Multiplicadores de esfuerzo en el Modelo Post-Arquitectura
 - Factores del producto
 - Factores de la plataforma

- Factores del personal
 - Factores del proyecto
 - Multiplicadores de esfuerzo en el Modelo de Diseño Inicial
- Cálculo del tiempo de desarrollo del proyecto
- Comparación entre COCOMO 81 y COCOMO II

LECCIÓN No. 6

PLANIFICACIÓN Y GESTIÓN DE PROYECTOS

- Identificar actividades o tareas
- Estimación de los tiempos y costes de las actividades
 - Distribución del tiempo
- Asignación de recursos
- Elaboración del plan
- Definición de hitos de control
- Calendarización
- Gestión de Proyectos
 - Control de un proyecto
 - Supervisión y revisión del proyecto
 - Acciones correctivas

LECCIÓN No. 7

GESTIÓN DE RIESGOS

- Administración de riesgos
 - Identificación del riesgo
 - Riesgos de tecnología
 - Riesgos de personas
 - Riesgos organizacionales
 - Riesgos de herramientas
 - Riesgos de requerimientos
 - Riesgos de estimación
 - Análisis de riesgos
 - Planeación de riesgos
 - Estrategias de anulación
 - Estrategias de disminución
 - Plan de contingencia
 - Estrategias de riesgo preactivas y reactivas.
 - Supervisión del riesgo
 - Plan RSGR

2.3 Desarrollo del Contenido de cada Lección

2.3.1 LECCIÓN No. 1: Conceptos, evolución y mitos de la Ingeniería de Software

Concepto

Este término fue introducido a finales de los 60 a raíz de la crisis del software.

La ingeniería de software es el establecimiento y uso de principios sólidos de ingeniería, orientados a obtener software económico que sea fiable y trabaje de manera eficiente en máquinas reales” (2.1)

Al software se debe considerar como un producto eficaz y barato tanto en el desarrollo como en la compra. También se requiere una serie de características como fiabilidad, facilidad de mantenimiento y de uso, eficiencia, etc.

El software es el producto que diseñan y construyen los ingenieros del software de cualquier tamaño y arquitectura.

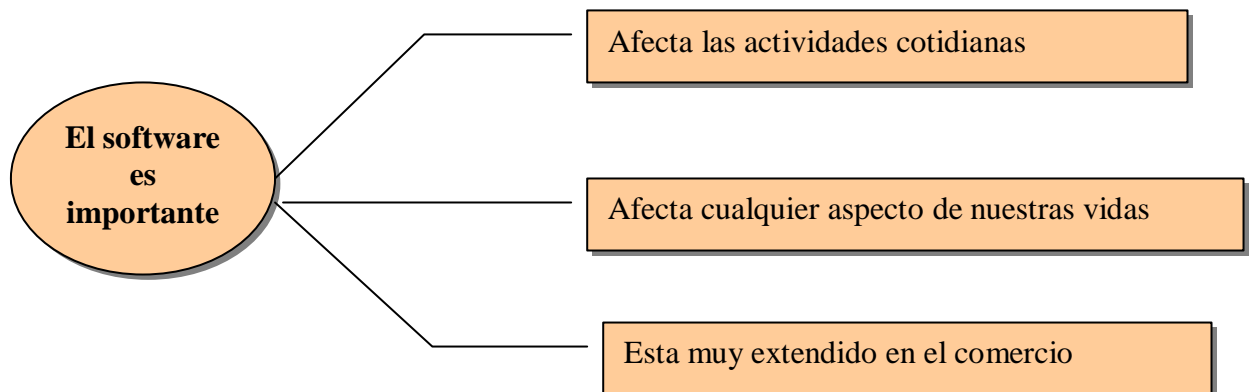


Figura 2.2: Relevancia del software

El producto obtenido
(Software)

(2.1) Fritz Bauer).

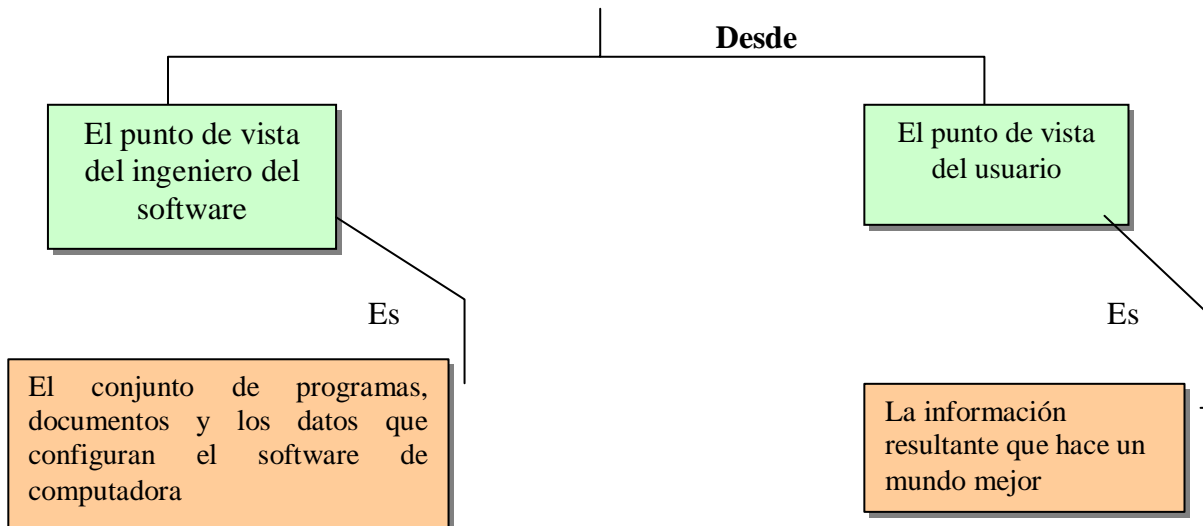


Figura 2.3: Producto obtenido

PROBLEMAS DEL SOFTWARE.

- La planificación y estimación de costos frecuentemente son imprecisas.
- Falta de “productividad” en la comunidad de software”
- La calidad del software es a veces ni aceptable.

Estos problemas al final crean insatisfacción y falta de confianza de los clientes. Los problemas anteriores son solo manifestación de otras dificultades:

- No tenemos tiempo de recoger datos sobre el proceso de desarrollo del software.
- Los proyectos de desarrollo de software se llevan a cabo con solo una vaga indicación de los requisitos del cliente.
- La calidad del software es normalmente cuestionable.
- El mantenimiento de software es muy costoso y no se le ha considerado un aspecto importante.

Crisis del software

La crisis se caracterizo por los siguientes problemas:

- Imprecisión en la planificación del proyecto y estimación de los costos.
- Baja calidad del software.
- Dificultad de mantenimiento de programas con un diseño poco estructurado, etc.

EVOLUCIÓN DEL SOFTWARE

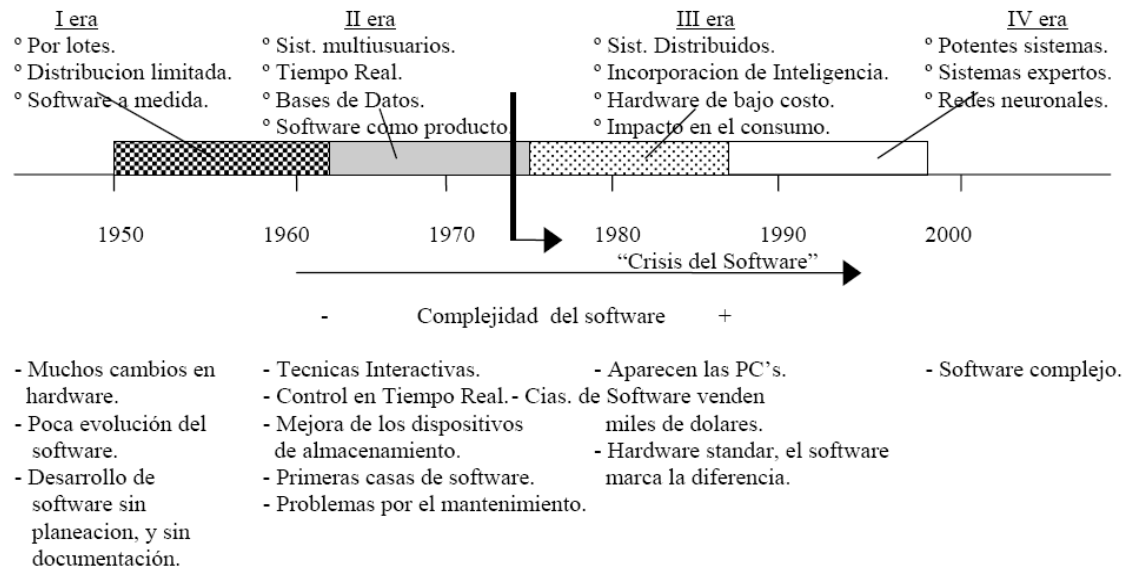


Figura 2.4: Evolución del software^(2.2)

La “Crisis del Software” se le conoce a una etapa en la que todos los programas desarrollados se corregían cuando había fallos o modificados a necesidades cambiantes, requerían de altos esfuerzos por mantenerlos, con mayor costo a medida que la complejidad del software crecía. En las pasadas décadas los ejecutivos y desarrolladores se hacían las siguientes preguntas: ¿Por qué lleva tanto tiempo terminar los programas? ¿Por qué es tan elevado el costo? ¿Por qué no podemos detectar los errores antes de entregar el software a los clientes? ¿Por qué resulta tan difícil constatar el progreso del desarrollo del software? Estas y otras preguntas manifiestan el carácter del software y la forma en que se desarrolla, estos problemas hacen necesaria la adopción de técnicas de Ingeniería de Software.

Mitos del Software

Los mitos del software tienen ciertos atributos que los convierten en insidiosos, así como se detalla en la siguiente tabla 2.1.

Tabla 2.1: Mitos del Software

	Mito: Ya se tiene un libro lleno de estándares y procedimientos
--	--

(2.2) Roger S. Pressman. Ingeniería de Software. Un enfoque practico. 4ta Edición

Mitos de la Administración o Gestión	para la construcción de software. ¿Esto proporcionará todo el conocimiento necesario?
	Realidad: Tal vez sea que el libro de estándares existe, pero no se lo usa o no responde a las interrogantes.
	Mito: Si se está atrasado en el itinerario es posible contratar más programadores para así terminar a tiempo.
	Realidad: Agregar gente a un proyecto de software atrasado lo atrasa más ^(2.3) .
Mitos del Cliente	Mito: Un enunciado general de los objetivos es suficiente para comenzar a escribir programas; los detalles se pueden afinar después.
	Realidad: Si el enunciado de los requerimientos es ambiguo puede ser llevar a un desastre. Los requerimientos precisos se los levanta mediante la comunicación continua y efectiva entre el cliente y el desarrollador.
	Mito: Los requerimientos del proyecto cambian de manera continua, pero el cambio puede ajustarse con facilidad porque el software es flexible.
	Realidad: Los requerimientos del software pueden cambiar, pero el impacto del cambio varía de acuerdo con el momento en que éste se introduce. Si el cambio de requerimiento se lo realiza conforme pasa el tiempo el costo de impacto es mayor provocando cambios en el diseño y recursos adicionales.
Mitos del Desarrollador	Mito: Una vez que el programa a sido escrito y puesto a funcionar, el trabajo está terminado.
	Realidad: Todo el esfuerzo aplicado en el software se realiza después de que el sistema haya sido entregado al cliente por primera vez, según datos de la industria.
	Mito: Mientras el programa no se esté ejecutando, no existe forma de evaluar su calidad.

(2.3 Brooks BRO75

	Realidad: Para asegurar una efectiva calidad del software se puede aplicar desde el inicio del proyecto: la revisión técnica formal.
--	---

Concepto de ingeniería de software

“Es una disciplina que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza” (2.4).

Objetivos

El objetivo de la Ingeniería del Software es la aplicación práctica y sistemática del conocimiento científico a:

La producción de programas correctos, que se desarrollan a tiempo y dentro de las estimaciones de presupuesto, a la correspondiente documentación para desarrollarlos, usarlos y mantenerlos. (2.5)

1. mejorar la calidad de los productos de software
2. aumentar la productividad y trabajo de los ingenieros del software.
3. Facilitar el control del proceso de desarrollo de software.
4. Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
5. Definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado.

¿Por que es importante ser ingeniero de software?

El Ingeniero de Software adopta un enfoque sistemático para llevar a cabo su trabajo y utilizan las herramientas y técnicas necesarias para resolver el problema planteado, de acuerdo a las restricciones de desarrollo y recursos disponibles.

(2.4) Sommerville (2002)

(2.5) Boehm

Los Ingenieros de software no solo deben considerar aspectos técnicos, deben tener una visión mas amplia, en lo ético, social y profesional.

Importancia de la Ingeniería de Software

La Ingeniería de software no solo concierne al desarrollo de sistemas o productos de software sino que promulga el uso de herramientas, métodos, técnicas para crear, mantener, producir y asegurar el producto o software con alta calidad, cumpliendo con los requerimientos del cliente logrando altos niveles de eficiencia, eficacia, flexibilidad entre otros.

El software es la clave del éxito de muchas empresas y este marca la diferencia de uno u otra, en la exactitud, suficiencia y oportunidad de la información generada por el software utilizado.

Ejemplo de importancia de la ingeniería de software.

Dos consultorios dentales, los dos cuentan con los últimos modelos de computadora personal y destinadas a apoyar las tareas y procesos relacionadas con el consultorio. Pero uno de ellos cuenta con un dispositivo especial conectado a la computadora y un software para obtener radiografías de piezas dentales, en un par de minutos la muestra radiográfica esta en pantalla y el profesional puede obtener diferentes vistas de la placa usando el software. Además puede establecer una conexión a través de internet o vía modem para enviar el archivo de la radiografía a otro colega experto con el fin de consultar y apoyar el diagnostico, todo esto en la misma cita.

La importancia recae en que la ingeniería de software analiza todos los problemas que puedan generarse, y este ataca todos los diferentes escenarios mediante en análisis de los procesos que puedan surgir.

Hace un par de décadas se sostenía la teoría de que los países que poseían los mejores recursos naturales estaban destinados a ser los mas ricos y poderosos del mundo, en México por ejemplo, se manejo la idea de que el petróleo era la puerta de entrada grande al mundo desarrollado.

El desarrollo de software se ha convertido en una industria con crecimiento vertical en los últimos años, hoy por hoy uno de los hombres más ricos del mundo son dueños de una casa de software (Microsoft, Google, yahoo, etc...)

Características del Software

Mantenibles.- debe ser posible que el software evolucione y que siga cumpliendo con sus especificaciones.

Confiabilidad.- el software no debe causar daños físicos o económicos en el caso de fallos.

Eficiencia.- el software no debe desperdiciar los recursos del sistema.

Utilización adecuada.- el software debe contar con una interfaz de usuario adecuada y su documentación.

Componentes de la ingeniería de Software

La ingeniería de software surge de la ingeniería de sistemas y de hardware. Abarca un conjunto de tres elementos que facilitan el control sobre el proceso de desarrollo de software y suministran las bases para construir software de calidad de una forma productiva:

- Métodos
- Herramientas
- Procedimientos

Métodos: indican cómo construir el software técnicamente e incluyen un amplio espectro de métodos para la planificación, la estimación, el análisis, el diseño, codificación, prueba y mantenimiento.

Ejemplos: las pruebas de corrección son método formales para verificación del software; el diseño orientado a objetos es un método de diseño; el COCOMO es un método de estimación de costos de producir software.

Herramientas: automáticas y semiautomáticas que apoyan a la aplicación de los métodos. Cuando se integran las herramientas de forma que la información creada por una herramienta puede ser usada por otra, se establece un sistema para el soporte del

desarrollo de software, llamado Ingeniería de Software Asistida por Computadora (CASE).

Procedimientos: que definen la secuencia en la que se aplican los métodos, herramientas, controles de calidad y guías para evaluación del progreso. ^(2.6)

Ejemplo: un ejemplo claro es la etapa de Diseño del ciclo de vida del Software, en éste se identificará el método de diseño y las herramientas para el desarrollo del sistema, que en conjunto forman un procedimiento.

Retos de la ingeniería de Software

Los retos que actualmente tienen la ingeniería es mantener los sistemas legados, desarrollar la diversidad de sistemas con más demanda y lograr que tiempos de entrega sean cada vez más pequeños.

Sistemas Legados.- Sistemas antiguos que deben ser mantenidos y mejorados, por que a lo largo de la vida de un producto este tiene a requerir cambios que mejoren e integren nuevas características del sistema inicial.

Heterogeneidad.- estos sistemas que incluyen una mezcla de software y hardware son productos que requieren necesariamente la interoperatividad de los dos, ya que estos cumplen funciones específicas complejas.

Entrega.- existe una presión incremental por una entrega a tiempo de los productos de software, que garanticen la entrega del producto en el tiempo estipulado.

Formalidad.- Existe una gran demanda de que exista formalidad en el proceso de desarrollo de software ^(2.7), esto nos garantiza de que el producto o software halla tenidos un transcurso o proceso conocido para se desarrollo.

Aplicaciones del Software

▪ Sistemas de tiempo real

Coordina, analiza y controla sucesos del mundo real conforme ocurre; manejando diferentes componentes.

(2.6) Adaptado de modulo I, Gestión de proyectos Software I, Unidad I, Universidad Politécnica De Madrid

(2.7) Adaptado de modulo I, Gestión de proyectos Software I, Unidad I, Universidad Politécnica De Madrid

Ejemplo: Sistemas médicos como los marcadores de pasos artificiales.

- **Sistemas de Gestión.**

Acceden a una o más bases de datos.

Ejemplo: Sistemas discretos de nóminas, cuentas de haberes/débitos, inventarios

- **Ingeniería y científico.**

Algoritmos que sirven para ejecutar una tarea o resolver un problema de manejo de números.

Ejemplo: Astronomía

- **Empotrado**

Controla productos y sistemas de mercados industriales y de consumo, ejecuta funciones muy limitadas.

Ejemplo: Sistemas de frenado.

- **Inteligencia artificial**

La IA es la ciencia que enfoca su estudio a lograr la comprensión de entidades inteligentes.

Ejemplo: Redes neuronales

- Computadores personales.

Aplicaciones que pertenecen a esta categoría:

- El procesamiento de textos
- Hojas de cálculo
- Los gráficos por computadora
- Multimedia
- Entretenimientos
- Gestión de bases de datos
- Redes o acceso a bases de datos externas

- **Basado en Web.**

Las páginas Web buscadas por un explorador son software que incorpora instrucciones ejecutables (por ejemplo, HTML, Perl, o Java), y datos (por ejemplo, hipertexto y una variedad de formatos de audio y visuales).

2.3.2 LECCIÓN No. 2: Modelos del Proceso de Software

Definición de un Modelo de Ciclo de Vida

Un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de software.

Un ciclo de vida debe:

- Determinar el orden de las fases del Proceso de software
- Establecer los criterios de transición para pasar de una fase a la siguiente.

No existe un modelo de ciclo de vida que sirva para cualquier proyecto. Cada proyecto debe seleccionar un ciclo de vida que sea el más adecuado para su caso. El ciclo de vida apropiado se elige en base a: la cultura de la corporación, el deseo de asumir los riesgos, el área de aplicación, la volatilidad de los requisitos, y hasta qué punto se entienden dichos requisitos. El ciclo de vida elegido ayuda a relacionar las tareas que forman el proceso de software de cada proyecto.

Etapas del Ciclo de Vida

Las etapas principales a realizar en cualquier ciclo de vida son:

1. **Análisis:** Construye un modelo de los requisitos
2. **Diseño:** A partir del modelo de análisis se deducen las estructuras de datos, la estructura en la que descompone el sistema y la interfaz de usuario.
3. **Codificación:** Construye el sistema. La salida de esta fase es código ejecutable.
4. **Pruebas:** Se comprueba que se cumplen criterios de corrección y calidad.
5. **Mantenimiento:** En esta fase, que tiene lugar después de la entrega se asegura que el sistema siga funcionando y adaptándose a nuevos requisitos.

MODELO DE CICLOS DE VIDA

Modelo en Cascada

Este modelo indica que el desarrollo de software puede ser a través de una secuencia simple de fases. Cada fase tiene un conjunto de metas bien definidas, y las actividades dentro de una fase contribuyen a la satisfacción de metas de esa fase.

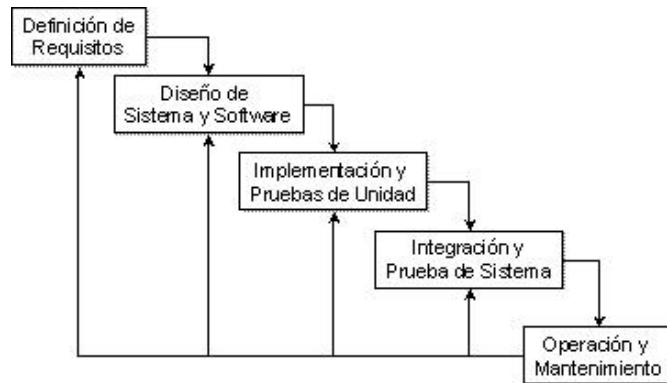


Figura 2.5: Modelo de desarrollo en cascada.

Características

- Las iteraciones son costosas e implican rehacer trabajo debido a la producción y aprobación de documentos.
- Aunque son pocas iteraciones, es normal congelar parte del desarrollo y continuar con las siguientes fases.
- Los problemas se dejan para su posterior resolución, lo que lleva a que estos sean ignorados o corregidos de una forma poco elegante.

Modelo De Desarrollo Evolutivo

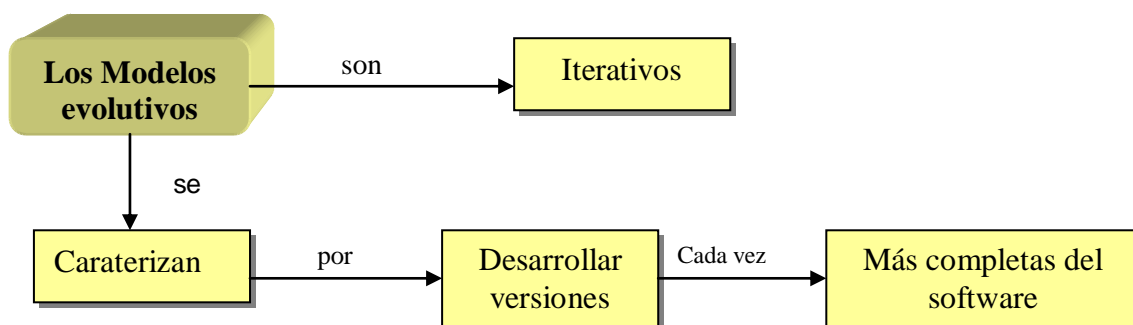


Figura 2.6: Modelos evolutivos

En el modelo evolutivo, los requerimientos son cuidadosamente examinados, y sólo esos que son bien comprendidos son seleccionados para el primer incremento. Los desarrolladores construyen una implementación parcial del sistema que recibe sólo estos requerimientos.

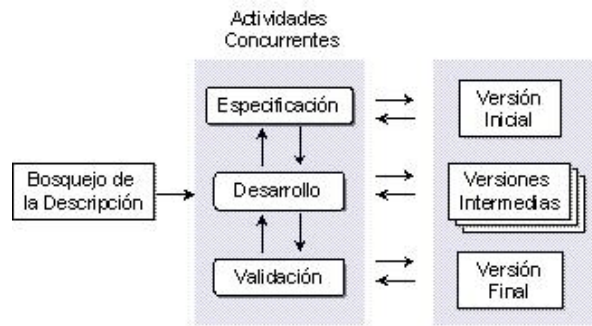


Figura 2.7: Modelo de desarrollo evolutivo.

Existen dos tipos de desarrollo evolutivo: Desarrollo Exploratorio y Enfoque utilizando prototipos.

Características

Entre los puntos favorables de este modelo están:

- La especificación puede desarrollarse de forma creciente.
- Los usuarios y desarrolladores logran un mejor entendimiento del sistema. Esto se refleja en una mejora de la calidad del software.
- Es más efectivo que el modelo de cascada, ya que cumple con las necesidades inmediatas del cliente.

Este modelo de desarrollo que, a diferencia de los prototipos, busca reemplazar el viejo sistemas con uno nuevo que tendría la propiedad de satisfacer los nuevos requerimientos el más rápido posible.

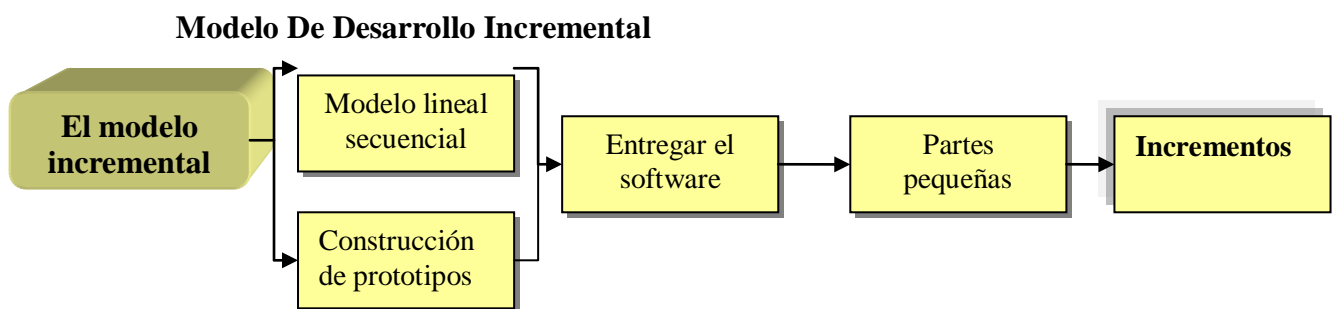


Figura 2.8: Modelo incremental

Sugirió^(2.8) el enfoque incremental de desarrollo como una forma de reducir la repetición del trabajo en el proceso de desarrollo y dar oportunidad de retrasar la toma de

(2.8) Mills

decisiones en los requisitos hasta adquirir experiencia con el sistema (ver Figura 2.5). Es una combinación del Modelo de Cascada y Modelo Evolutivo.

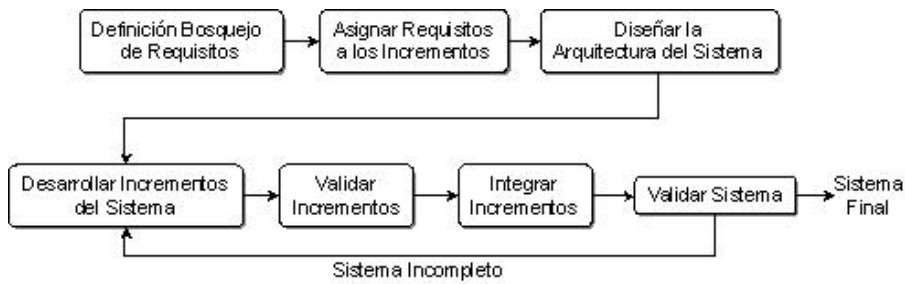


Figura 2.9: Modelo de desarrollo iterativo incremental.

Características

- Construir un sistema pequeño es siempre menos riesgoso que construir un sistema grande.
- Al ir desarrollando parte de las funcionalidades, es más fácil determinar si los requerimientos planeados para los niveles subsiguientes son correctos.
- Si un error importante es realizado, sólo la última iteración necesita ser descartada.

Modelo Espiral

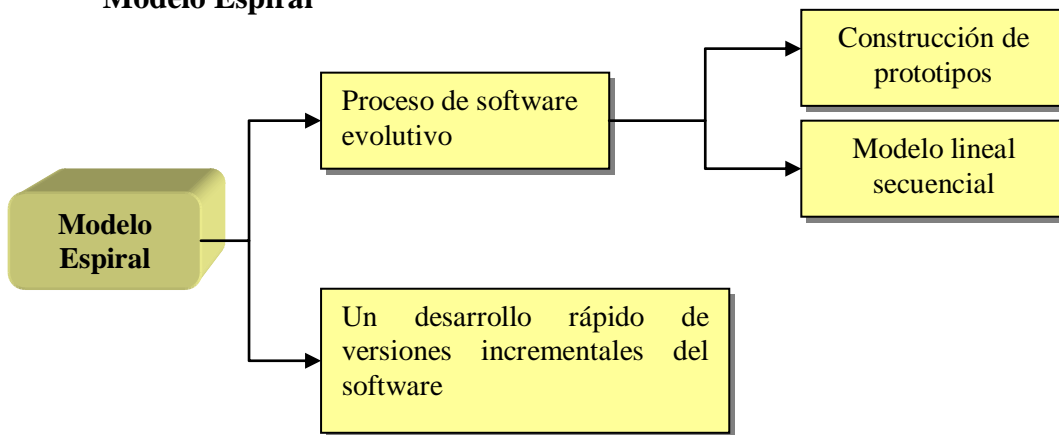


Figura 2.10: Modelo espiral

En este modelo, el esfuerzo de desarrollo es iterativo. Tan pronto como uno completa un esfuerzo de desarrollo, otro comienza. Además, en cada desarrollo ejecutado existen especificaciones a seguir

Este modelo a diferencia de los otros toma en consideración explícitamente el riesgo, esta es una actividad importante en la administración del proyecto.

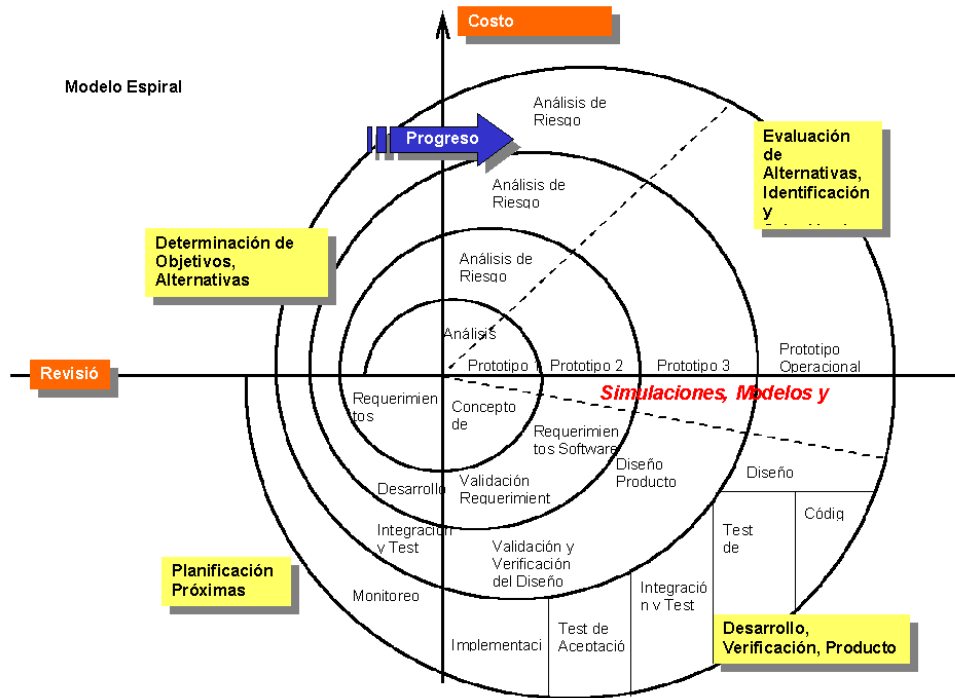
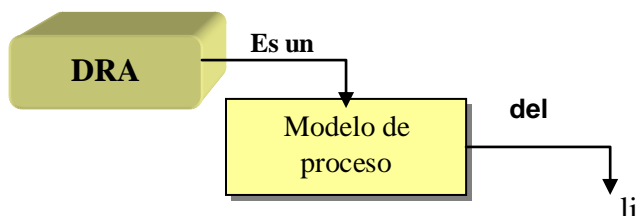


Figura 2.11: Modelo de desarrollo en Espiral

Características

- Decidir qué problema se quiere resolver antes de viajar a resolverlo.
- Examinar tus múltiples alternativas de acción y elegir una de las más convenientes.
- Evaluar qué tienes hecho y qué tienes que haber aprendido después de hacer algo.

Modelo DRA (Desarrollo Rápido de Aplicación)



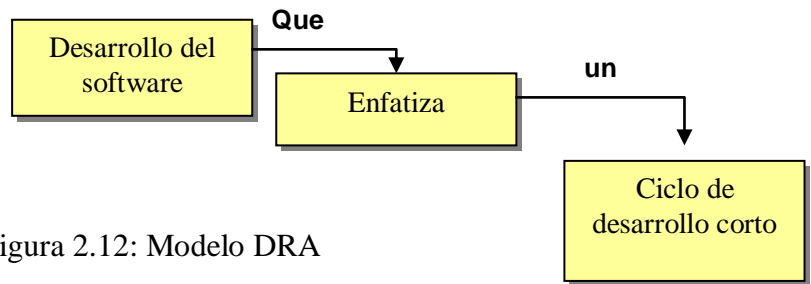


Figura 2.12: Modelo DRA

El Modelo DRA consiste en un desarrollo rápido de aplicaciones basado en el modelo lineal secuencial, pero donde se enfatiza un ciclo de desarrollo extremadamente corto.

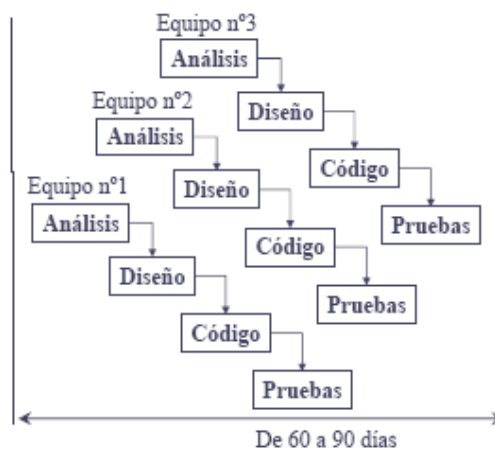


Figura 2.13: Modelo de desarrollo DRA

Inconvenientes

- Los clientes y desarrolladores deben estar comprometidos en las rápidas actividades.
- No es adecuado con riesgos técnicos altos
- Para proyectos grandes necesitamos de recursos suficientes para formar los equipos necesarios.

Modelo de Prototipos

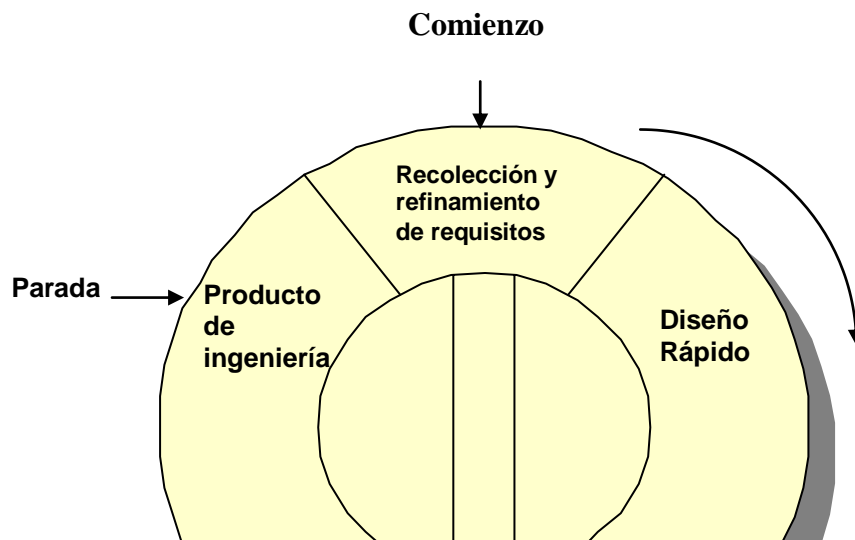




Figura 2.14 Modelo de Prototipo

A veces suele ocurrir en los proyectos de software, el usuario no tiene ideas de lo que necesita, o que el ingeniero de software no está segura de la viabilidad de la solución que se tiene en mente. Estas condiciones podrían ser las adecuadas o por lo menos un acercamiento para la realización de un prototipo.

Características

- No modifica el flujo del ciclo de vida.
- Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios.
- Reduce costos y aumenta la probabilidad de éxito.
- Exige de exponer de las herramientas adecuadas.
- No presenta calidad ni robustez.
- Una vez identificados todos los requisitos mediante el prototipo, se construye el producto de ingeniería.

Un problema del prototipo es la elección de las funciones que se desean incorporar, y cuales son las que hay que dejar afuera, pues se corre el riesgo de incorporar características no relevantes.

¿Cuál es el modelo de proceso más adecuado?

Cada proyecto de software requiere de una forma de particular de abordar el problema. En la Tabla 2.2 se expone un cuadro comparativo de acuerdo con algunos criterios básicos para la selección de un modelo de proceso, la medida utilizada indica el nivel de efectividad del modelo de proceso de acuerdo al criterio.

Tabla 2.2: Comparación entre modelos de proceso de software

Modelo de proceso / Criterio	Desempeño si no se predefinen requisitos	Produce software fiable	Gestión de riesgos	Permite cambio sobre la marcha	Visibilidad del progreso
Desarrollo en cascada	<i>Bajo</i>	<i>Alto</i>	<i>Bajo</i>	<i>Bajo</i>	<i>Bajo</i>
Desarrollo Evolutivo Y Prototipado	<i>Alto</i>	<i>Medio</i>	<i>Medio</i>	<i>Alto</i>	<i>Alto</i>
Desarrollo Incremental	<i>Bajo</i>	<i>Alto</i>	<i>Medio</i>	<i>Medio</i>	<i>Bajo</i>
Desarrollo Espiral	<i>Alto</i>	<i>Alto</i>	<i>Alto</i>	<i>Medio</i>	<i>Medio</i>

Modelos de Proceso de software

¿Qué es un proceso software?

Es un conjunto de actividades y resultados asociados que producen un producto de software. Es uno de los componentes de un método de desarrollo de software.^(2.9)

Distintos procesos de software organizan las actividades de diferentes formas, y las describen con diferente nivel de detalle. El tiempo de cada actividad varía, así como los

(2.9) (Sommerville 2005)

resultados. Organizaciones diferentes usan procesos diferentes para producir el mismo producto. Sin embargo, para algunos tipos de aplicación, algunos procesos son más convenientes que otros.

¿Qué es un modelo de procesos del software?

Es una descripción de un proceso de software que se presenta desde una perspectiva particular. ^(2.10)

- Es una abstracción de un proceso real.
- Incluye actividades (que son parte de los procesos del software), los productos software, y el papel de las personas interesadas en el desarrollo (stakeholders).

Estos modelos, pueden ser:

- Un modelo de flujo de trabajo (workflow)
- Un modelo de flujo de datos o de actividad
- Un modelo de rol/acción

Una perspectiva utilizada para determinar los elementos del proceso de desarrollo de software es establecer las relaciones entre elementos que permitan responder Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.

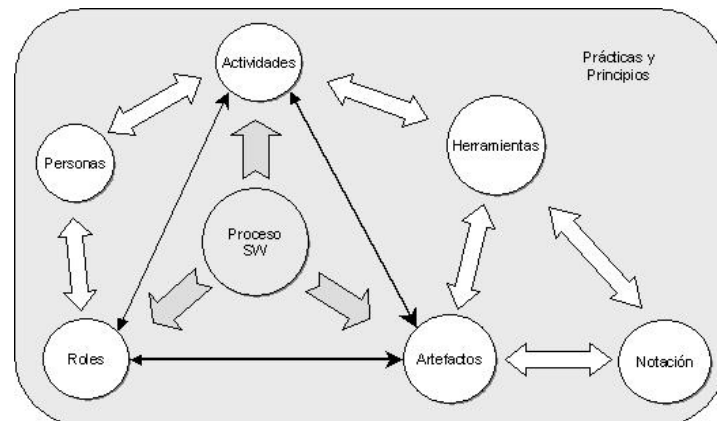


Figura 2.15: Relación entre elementos del proceso del software

En la Figura 2.33 se muestran los elementos de un proceso de desarrollo de software y sus relaciones. Así las interrogantes se responden de la siguiente forma:

- **Quién:** Las Personas participantes en el proyecto de desarrollo desempeñando uno o más Roles específicos.

(2.10) (Sommerville 2005)

- **Qué:** Un Artefacto^(2.11) es producido por un Rol en una de sus Actividades. Los Artefactos se especifican utilizando Notaciones específicas. Las Herramientas apoyan la elaboración de Artefactos soportando ciertas Notaciones.
- **Cómo y Cuándo:** Las Actividades son una serie de pasos que lleva a cabo un Rol durante el proceso de desarrollo. El avance del proyecto está controlado mediante hitos que establecen un determinado estado de terminación de ciertos Artefactos.

Metodologías Basadas Estándar IEEE sobre Proceso de Software

El estándar IEEE 1074[1989] describen en detalle las fases o subprocesos que conforman el Proceso Base de Construcción del software, cada subproceso se detalla a nivel de propósito, actividades involucradas y documentación principal propuesta por el estándar.

El proceso software está compuesto, de cuatro procesos principales, cada uno de los cuales agrupa una serie de actividades que se encargan de la realización de sus requisitos asociados. Estos procesos son los siguientes:

- Proceso de selección de un Modelo de Ciclo de Vida del Producto
- Procesos de Gestión del Proyecto
- Procesos Orientados al Desarrollo del Software
- Procesos Integrales del proyecto

Proceso de selección de un Modelo de Ciclo de Vida del Producto

Durante este proceso se selecciona un ciclo de vida que establece el orden de ejecución de las distintas actividades, marcadas por el proceso e involucradas en el proyecto.

Actividades a realizar:

Identificar los posibles modelos de ciclo de vida del software

Seleccionar el modelo más adecuado para el proyecto

Documento de salida:

(2.11) Un artefacto es una pieza de información que (1) es producida, modificada o usada por el proceso, (2) define un área de responsabilidad para un rol y (3) está sujeta a control de versiones. Un artefacto puede ser un modelo, un elemento de modelo o un documento.

PROCESOS DE GESTIÓN DEL PROYECTO

Crea la estructura del proyecto y asegura el nivel apropiado de gestión del mismo durante todo el ciclo de vida del software. Involucra actividades de planificación, estimación de recursos, seguimiento y control, y evaluación del proyecto.

Proceso de Iniciación del Proyecto.- Abarca aquellas actividades de creación de la estructura del proyecto.

Actividades a realizar:

Establecer el mapa de actividades para el modelo de ciclo de vida del software seleccionado

Asignar los recursos del proyecto

Definir el entorno del proyecto

Planificar la gestión del proyecto

Documento de salida:

Plan de Gestión del Proyecto

Plan de Retiro

Proceso de seguimiento y control del Proyecto.- En este proceso se realiza un análisis de riesgos (técnico, económico, operativo y de soporte, y del programa o calendario) que permiten identificar los problemas potenciales.

Actividades a realizar:

Analizar riesgos

Realizar la planificación de contingencias

Gestionar el proyecto

Archivar registros

Documento de salida:

Análisis de riesgos

Plan de contingencias

Proceso de Gestión de la calidad del Software.- Su objetivo es la planificación y administración de las acciones necesarias para proveer una confianza adecuada en la calidad de los productos software.

Actividades a realizar:

Planificar la garantía de la calidad del software

Desarrollar métricas de calidad

Gestionar la calidad del software

Identificar necesidades de mejora de la calidad

Documento de salida:

Plan de garantía de calidad del software

Recomendaciones de mejora de calidad software

PROCESOS ORIENTADOS AL DESARROLLO DEL SOFTWARE

Producen, instalan, operan y mantienen el software y lo retiran de su uso. Se clasifican en procesos de pre-desarrollo, desarrollo y post-desarrollo.

Procesos de Pre-Desarrollo

Son los procesos que se deben realizar antes de que comience el desarrollo de software.

Proceso de Exploración de Conceptos.- Este proceso incluye la identificación de una necesidad, la formulación de soluciones potenciales, su evaluación y refinamiento a nivel del sistema.

Actividades a realizar:

Identificar ideas y necesidades

Formular soluciones potenciales

Conducir estudios de viabilidad

Planificar la transición del sistema (si se aplica)

Documento de salida:

Modelo de la situación actual

Informe preliminar de necesidades

Proceso de Asignación del Sistema.- Este proceso se realiza cuando el sistema requiere tanto del desarrollo de hardware y software.

Actividades a realizar:

Analizar las funciones del sistema

Desarrollar la arquitectura del sistema

Descomponer los requisitos del sistema

Documento de salida:

Especificación de requisitos funcionales del software

Especificación de requisitos funcionales del hardware

Especificación de la interfaz del sistema

Descripción funcional del sistema

Arquitectura del Sistema

Procesos de Desarrollo

Son los procesos que se deben realizar para la construcción del producto de software. Estos definirán qué información obtener y cómo estructurar los datos, qué algoritmos usar para procesar los datos y cómo implementarlos y qué interfaces desarrollar para operar con el software y cómo hacerlo.

Proceso de Requisitos.- Incluye las actividades iterativas dirigidas al desarrollo de la Especificación de Requisitos del Software. Para la determinación completa y consistente de los requisitos del software el análisis se enfatiza sobre la salida resultante, la descomposición de los datos, el procesamiento de los datos, las bases de datos (si existen) y las interfaces del usuario, del software y del hardware.

Actividades a realizar:

Definir y desarrollar los requisitos del software

Definir los requisitos de interfaz

Priorizar e integrar los requisitos del software

Documento de salida:

Especificación de requisitos del software

Requisitos del interfaz con el usuario

Requisitos del interfaz con otro software

Requisitos del interfaz con el hardware

Requisitos del interfaz con el sistema físico

Proceso de diseño.- Es el proceso central que unifica los procesos de desarrollo y de mantenimiento del software. Su objetivo es desarrollar una representación coherente y organizada del sistema software que satisfaga la Especificación de Requisitos del Software.

Actividades a realizar:

Realizar el diseño arquitectónico

Analizar el flujo de información

Diseñar la base de datos (si se aplica)

Diseñar los interfaces

Seleccionar o Desarrollar algoritmos (si se aplica)

Realizar el diseño detallado

Documento de salida:

Descripción de diseño del software

Descripción de la arquitectura del software

Descripción del flujo de información

Descripción de la base de datos

Descripción de las interfaces

Descripción de los algoritmos

Proceso de implementación.- Este proceso transforma la representación del diseño detallado de un producto software a una realización en un lenguaje de programación apropiado.

Actividades a realizar:

Crear los datos de prueba

Crear el código fuente

Generar el código objeto

Crear la documentación de operación

Planificar la integración

Realizar la integración

Documento de salida:

Datos para las pruebas

Documentación del sistema

Documentación de usuario

Plan de integración

Sistema software integrado

Procesos de Post-Desarrollo

Son los procesos que se deben realizar para instalar, operar, soportar, mantener y retirar un producto software. Se realizan después de la construcción del software, es decir, se aplican a las últimas fases del ciclo de vida del software.

Proceso de instalación.- Implica el transporte y la instalación de un sistema software desde el entorno de desarrollo al entorno destino.

Actividades a realizar:

Planificar la instalación

Distribuir el software

Instalar el software

Cargar la base de datos (si se aplica)

Aceptar el software en el entorno de operación

Realizar las actualizaciones (instalar el software probado)

Documento de salida:

Plan de instalación

Informe de instalación

Proceso de operación y soporte.- Involucra la operación del sistema por parte del usuario y el soporte continuo.

Actividades a realizar:

Operar el sistema

Proveer la asistencia técnica y consultas

Mantener el histórico de peticiones de soporte

Documento de salida:

Histórico de peticiones de soporte

Proceso de mantenimiento.- Este proceso se interesa por los errores, defectos, fallos, mejoras y cambios del software.

Actividades a realizar:

Reaplicar el ciclo de vida del software

Documento de salida:

Orden de mantenimiento

Recomendaciones de mantenimiento

Proceso de retiro.- Es la jubilación de un sistema existente de su soporte activo o de su uso mediante el cese de operación o soporte, o mediante su reemplazamiento.

Actividades a realizar:

Notificar al usuario

Conducir operaciones en paralelo (si se aplica)

Retirar el usuario

Documento de salida:

Plan de retiro

PROCESOS INTEGRALES DEL PROYECTO

Son procesos simultáneos y complementarios a los procesos orientados al desarrollo. Son necesarios para complementar con éxito las actividades del proyecto de software. Aseguran la terminación y calidad de las funciones del mismo.

Proceso de verificación y validación.- Abarca la planificación y la realización de todas las tareas de verificación.

Actividades a realizar:

Planificar la verificación y validación

Ejecutar las tareas de verificación y validación

Recoger y analizar los datos de las métricas

Planificar las pruebas

Desarrollar las especificaciones de las pruebas

Ejecutar las pruebas

Documento de salida:

Plan de verificación y validación

Informes de evaluación

Plan de pruebas

Especificación de las pruebas

Informe resumen de las pruebas

Software probado

Proceso de Gestión de la Configuración.- Este proceso involucra un conjunto de actividades desarrolladas para gestionar los cambios durante todo el ciclo de vida del software.

Actividades a realizar:

Planificar la gestión de la configuración

Realizar la identificación de la configuración

Realizar el control de la configuración

Realizar la información del estado de la configuración

Documento de salida:

Plan de gestión de configuración del software

Orden de cambio de ingeniería

Cambio de estado

Informe de estado

Proceso de desarrollo de documentación.- Es el conjunto de actividades que planifican, diseñan, implementan, editan, producen, distribuyen y mantienen los documentos necesarios para los desarrolladores y los usuarios.

Actividades a realizar:

Planificar la documentación

Implementar la documentación

Producir y distribuir la documentación

Documento de salida:

Plan de documentación

Proceso de formación.- Incluye la planificación, desarrollo, validación e implementación de los programas de formación de desarrolladores, personal de soporte técnico y clientes y la elaboración de los materiales de formación adecuados.

Actividades a realizar:

Planificar el programa de formación

Desarrollar los materiales de formación

Validar el programa de formación

Implementar el programa de formación

Documento de salida:

Plan de formación

¿Diferencia entre ciclo de vida y procesos de software?

El proceso de desarrollo de software es un conjunto de macro procesos que contiene acciones como el ciclo de vida, gestión de proyectos, control y calidad, etc, con el fin de encontrar las necesidades del cliente y transformarlas en requerimientos bien definidos, en cambio el ciclo de vida es un paso que pertenece al proceso de desarrollo de software, que contiene un grupo de actividades ordenadas como son: requerimientos, análisis, diseño, codificación, pruebas y mantenimiento, actividades seleccionadas de acuerdo al problema a solucionar y su entorno.

2.3.3 LECCIÓN No. 3: Fundamentos de Gestión de Proyectos

Gestión de Proyectos

La gestión de proyectos de desarrollo de software se basa en los siguientes aspectos que muestra la figura 2.34

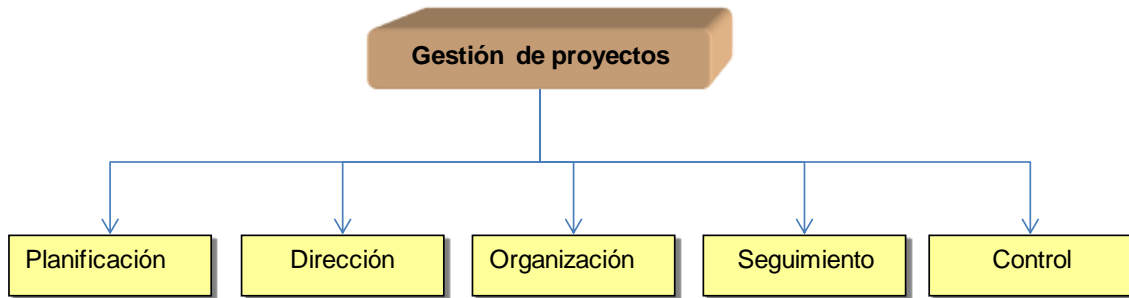


Figura 2.16: Gestión de proyecto de software

Antes de comenzar con cada uno de los aspectos que determina la toma de decisiones sobre la información de proyectos de software, aprenderemos sobre el espectro de gestión.

El espectro de la gestión

La gestión de proyectos de software se enfoca sobre las 4 P: personal, producto, proceso y proyecto.

El orden no es arbitrario. El gestor que olvida que el trabajo de ingeniería de software es una empresa inmensamente humana nunca tendrá éxito en la gestión de proyectos.

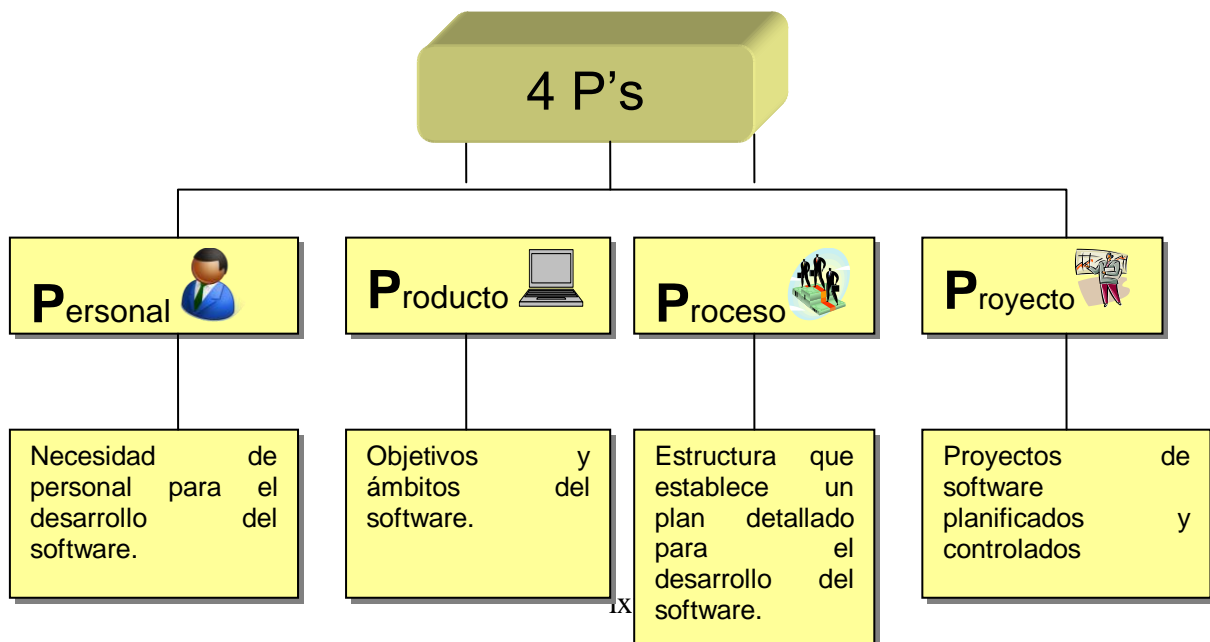


Figura 2.17: Espectro de Gestión

El Personal

La Cantidad de personas requeridas para el desarrollo de un proyecto de software solo puede ser determinado después de hacer una estimación del esfuerzo de desarrollo (por ejemplo personas mes o personas años), y seleccionar la posición dentro de la organización y la especialidad que desempeñara cada profesional.

La constitución del equipo de trabajo es la actividad más delicada con la que se enfrenta un Director de Proyecto, y en la que más debe demostrar sus capacidades.

En el momento de la selección del personal, los perfiles son analizados en base a una serie de características importantes a la hora de delimitar su participación en el proyecto:

- Responsabilidades o funciones a desempeñar en cada uno de los procesos.
- Perfil o características propias de cada uno de los participantes.

Tabla 2.3 Características del Personal

Perfiles	Responsabilidades
Perfil Directivo	<ul style="list-style-type: none"> • Personas con un nivel alto en la dirección de la organización. • Tener conocimiento de los objetivos estratégicos y de negocio que se persiguen. • Autoridad para validar y aprobar cada uno de los procesos realizados durante el desarrollo del Sistema de Información.
Perfil Jefe de Proyecto	<ul style="list-style-type: none"> • Realizar la estimación del esfuerzo necesario para llevar a cabo el proyecto. • Seleccionar la estrategia de desarrollo y estructura del mismo seleccionando los procesos principales que lo integran. • Fijar el calendario de hitos y entregas.

	<ul style="list-style-type: none"> • Establecer la planificación del proyecto.
Perfil del Analista	<ul style="list-style-type: none"> • Elaborar un catálogo detallado de requisitos que permita describir con precisión el sistema de información.
Perfil del Programador	<ul style="list-style-type: none"> • Construir el código que dará lugar al producto resultante en base al diseño técnico realizado por el analista, generando también el código asociado a los procedimientos de migración y carga inicial de datos.

El producto

Antes de planear un proyecto se deberían establecer los Objetivos y el ámbito del producto, considerar soluciones alternativas e identificar las restricciones técnicas y de gestión, para lo cual se deben reunir el desarrollador del software y el cliente.

Los *objetivos* identifican las metas *globales* del producto sin considerar como se lograrán.

El *ámbito* identifica los datos primarios, las funciones y los comportamientos que caracterizan al producto.

Ámbito del Software

El ámbito del proyecto no debe ser ambiguo ni comprensible a niveles de gestión y técnico. Se debe acotar un enunciado del ámbito del software, por ejemplo: establecer de manera explícita los datos cuantitativos, restricciones o limitaciones, describir los factores que reducen riesgos, etc.

Contexto.- ¿Cómo se encaja el software a construir en un sistema, producto o contexto de negocios mayor y que limitaciones se imponen como resultado del contexto?

Objetivos de la información.- ¿Qué objetos de datos visibles al cliente se obtienen del software? ¿Que objeto da datos son requeridos de entrada?

Función y rendimiento.- ¿Que función realiza el software para transformar la información de entrada en una salida? ¿hay características de rendimiento especiales que aborda?

El Proceso

El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de logra un objetivo, en este caso, la obtención de un producto de software de calidad".^(2.12) El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo.

Los pasos de un proceso de software son:

- **Requerimientos:** Se revisan los requerimientos necesarios para satisfacer las necesidades de los diferentes usuarios.
- **Análisis:** Se realiza un análisis de los requerimientos y se determinan las clases y algoritmos que se construirán.
- **Diseño:** Se determinan las técnicas que se utilizarán para implementar las clases y los algoritmos. Además se toman decisiones para obtener la mejor portabilidad y eficiencia del software.
- **Implementación:** Se realiza la escritura del código de acuerdo con el análisis y diseño hecho antes.
- **Pruebas:** Se realizan pruebas preliminares de diferentes partes del código.
- **Documentación:** En esta etapa se hace la documentación de lo realizado en cada iteración.

El proyecto

En éste punto se debe gestionar la complejidad del proyecto software y planificar, supervisar y controlar su gestión eficazmente. En la lección 6 se enfatizará la administración de proyectos.

(2.12) (Jacobson 1998).

El término proyecto se utiliza generalmente para referirse a todas las etapas que intervienen para poner en marcha el proyecto software. Así, la idea es definir con precisión la naturaleza del proyecto de manera que se puedan finalizar los documentos contractuales (preparación de un contrato) que permitan el compromiso entre el contratista y el cliente para poner en marcha el proyecto. En esta fase se formaliza la decisión de comenzar con el proyecto. Seguidamente se realizará una planificación temporal, identificando tareas, asignando recursos. La ejecución del proyecto se llevará a cabo después de establecer un calendario con sus respectivos hitos y terminando este ciclo con el seguimiento y control, si se ameritara se establecerán acciones correctivas.

Métricas del software

Importancia de medir

La medición de los procesos del software no es una tarea trivial, así para poder medir los procesos de forma efectiva desde el punto de vista de la medición hay que distinguir las entidades relevantes de forma aislada, integrar y relacionar la información para obtener resultados.

Dentro de las razones por las cuales se debe medir un proyecto de software se encuentra:

- Para indicar la calidad del producto
- Para evaluar la productividad de la gente que desarrolla el producto.
- Para evaluar los beneficios en términos de calidad, derivados del uso de nuevos métodos y herramientas de la ingeniería de software.
- Para establecer una línea de base para la estimación.
- Para ayudar a justificar el uso de nuevas herramientas o de formación adicional.
- Para establecer la confiabilidad (errores)
- Para medir la flexibilidad (adaptable a nuevas situaciones).
- Para medir su mantenibilidad (fácil de hacerle cambios).
- Para establecer su comprensión (Es entendible el código).
- Para hacer seguimiento y control, gestionar un proyecto de software

¿Qué son las métricas de software?

Se define a las métricas de software como “La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos para suministrar información relevante a tiempo, así el administrador junto con el empleo de estas técnicas mejorará el proceso y sus productos” (2.13). Las métricas de software proveen la información necesaria para la toma de decisiones técnicas. En la figura 2.18 se ilustra una extensión de esta definición que incluye los servicios relacionados al software como la respuesta a los resultados del cliente.

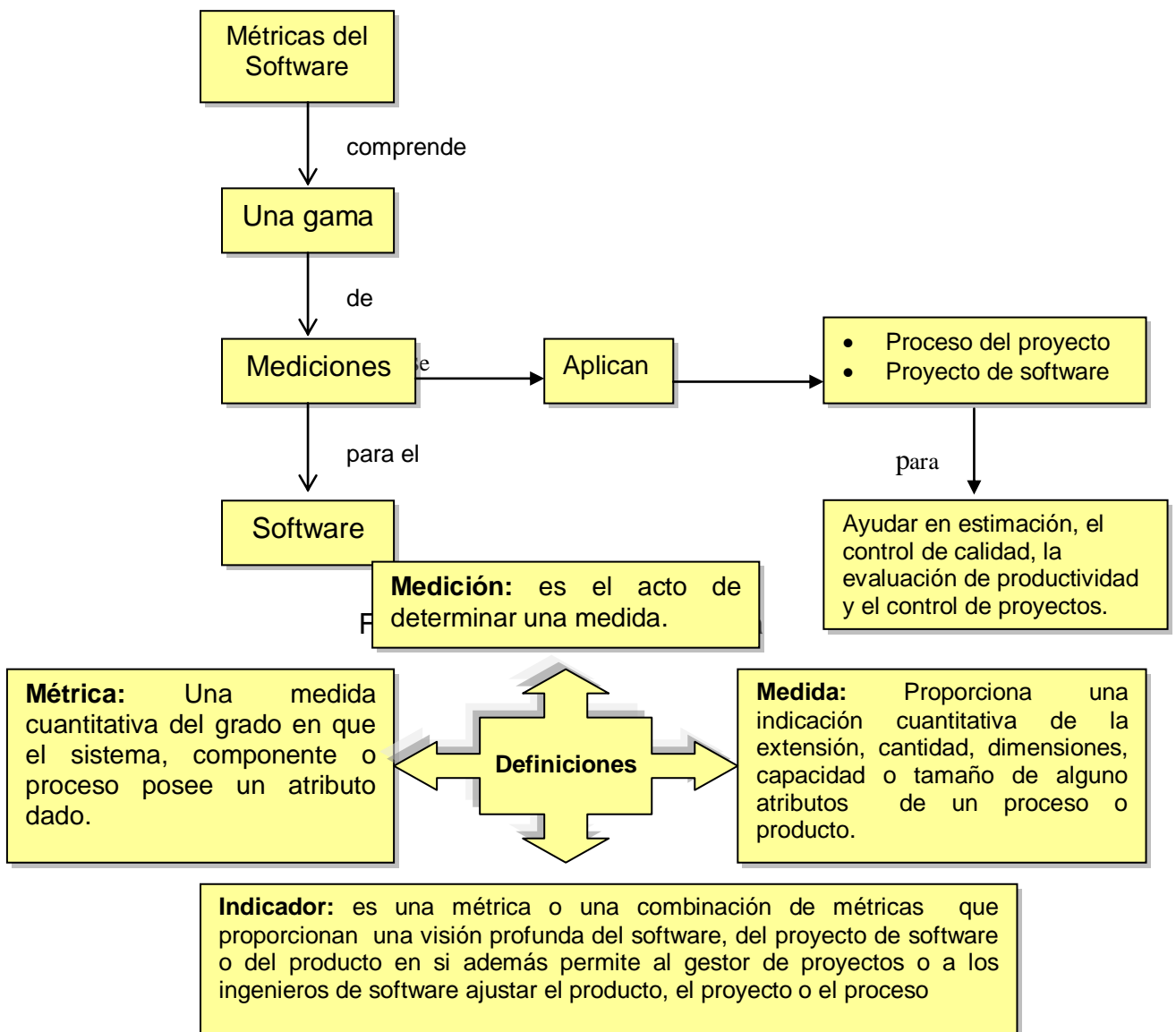


Figura 2.18: Servicios relacionados al Software

Las métricas deben responder a las siguientes preguntas: (2.14)

(2.13) (Michael M. '99).

(2.14) Adaptado de Fenton '91

- ¿Cuánto mide? - la complejidad en la medida
- ¿Qué tan bien mide? - la calidad en la medida
- ¿Qué tanto tiempo mide? - la predicción

Las métricas de software incluyen actividades, como:

- Estimación de costo y el esfuerzo
- Medición de la productividad
- Acumulación de datos
- Realización de modelos y mediciones de la calidad
 - Elaboración de modelos de seguridad
 - Evaluación y modelos de desempeño
 - Valoración de las capacidades y de la madurez
 - Administración por métricas
 - Evaluación del método y herramientas

Las funciones más importantes de las métricas

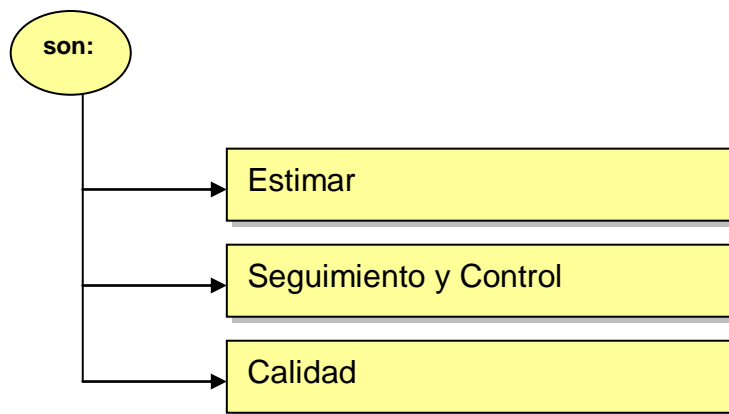


Figura 2.19: Funciones de las métricas

Características de las métricas

Tabla 2.4: Características de las métricas

Característica	Descripción
Simple y fácil de calcular	Su cálculo no requiere de un esfuerzo o a una cantidad de

	tiempo exagerado
Empírica e intuitivamente persuasiva	Satisface las nociones intuitivas del ingeniero de software sobre el atributo del producto en cuestión.
Consistente en el empleo de unidades y tamaños	El cálculo matemático de la métrica utiliza medidas que no lleven a extrañas combinaciones de unidades
Independiente del lenguaje de programación	Las métricas se apoyan en el modelo de análisis, modelo de diseño o en la propia estructura del programa.
Realimentación de calidad	La métrica suministra al desarrollador de software información que le llevará a un producto final de calidad superior.

Beneficios

- Provee una disciplina sólida que asegure una métrica bien definida basándose en las metas del cliente.
- Eliminar malentendidos acerca de la intención del empleo de la métrica.
- Comunicar la necesidad de la métrica, la cual puede ayudar en la obtención de recursos para la implantación de los mecanismos de colección y reporte de datos.
- Provee la base para el informe de requerimientos, para realizar un diseño eficiente de la métrica.

Métricas de Producto

Las métricas de producto pueden medir la complejidad del *diseño*, el *tamaño* del producto final (fuente u objeto) o el número de páginas de documentación producida.

Atributos internos del producto:

- Medidas de *tamaño* (longitud del código, Funcionalidad...)
- Medidas de *diseño*: grado de interdependencia entre módulos, grado en el los componentes locales de un módulo colaboran para realizar una tarea concreta.
- Medidas de *complejidad* (estructuras de datos, estructura, lógica...)

Atributos externos, que dependen del comportamiento del producto en un entorno determinado:

- Calidad

- Fiabilidad
- Usabilidad
- Facilidad de mantenimiento.

Métricas orientadas al tamaño

Esta métrica proviene de la normalización de las medidas y/o productividad considerando el tamaño del software que se haya producido. La métrica mas utilizada es *líneas de código* (LDC), tienen el inconveniente obvio de que sus valores no pueden ser medidos hasta que el proceso de codificación ha finalizado.

Los datos que se debe tener en cuenta, se pueden llevar en la siguiente tabla:

Proyecto	LDC	Esfuerzo	Costo \$	Pág. de documentación	Errores	Defectos	Personas
Neptuno	15.657	20	150.000	755	126	76	4
Keko	20.543	25	234.648	1234	120	86	5

Figura 2.20: Resultados de métricas orientadas al tamaño

Se selecciona las LDC como medida de normalización y se desarrolla métricas orientadas al tamaño:

Errores por KLDC (miles de líneas de código), defectos por KLDC, \$ por LDC, paginas de documentación por KLDC, errores/personas-mes, LDC personas-mes, \$/ página de doc.

La medida más utilizada para identificar la longitud del código fuente de un programa es el número de líneas de código (Lines of Code en ingles) sin embargo, esta métrica puede calcularse de varias maneras. Una **línea de código** es cualquier línea de un texto de un programa que no es un comentario o línea en blanco, sin tener en cuenta el número de instrucciones o parte de instrucciones en la línea.

Esta definición incluye todas las líneas que contienen cabecera de programas, declaraciones e instrucciones ejecutables y no ejecutables. Esta medida se suele representar por NCLOC (No comentary lines of code). Como puede verse no es una medida que refleje la longitud real de un programa. Su justificación está en el uso que

se ha hecho de ella en ciertos modelos para determinar el esfuerzo desde el punto de vista de evaluar la profundidad, sin embargo, si queremos conocer la longitud real del programa esta sería.

$LOC = NCLOC + CLOC$

Donde CLOC (Comentary lines of code) es el número de líneas de comentario

Las ventajas de éste tipo de métrica es su calculo fácil, existencia de modelos de estimación que usan LDC como entrada, existe documentación, etc.

Métricas de Proceso

Son medidas del proceso de desarrollo del software tales como el tiempo de desarrollo total, esfuerzo en días/hombre de desarrollo del producto, tipo de metodología utilizada o nivel medio de los programadores.

Los *Indicadores* del Proceso permiten obtener información sobre la eficacia de un proceso existente. Evaluar lo que funciona y lo que no funciona del proceso.

Las Métricas del Proceso se recopilan de proyectos anteriores, y su meta es lograr indicadores para mejorar los procesos del software a largo plazo.

Los aspectos relacionados con el proceso de desarrollo de software que pueden medirse son:

Atributos internos que pueden medirse directamente:

- Duración de un proceso o de una de sus actividades.
- Esfuerzo asociado con el proceso o con una de sus actividades.
- Número de incidentes de un tipo determinado que ocurren durante el proceso o una de sus actividades.

Esas medidas pueden combinarse con otras para tener un mayor conocimiento del proceso de desarrollo.

- Atributos externos del Proceso:

- Cambios durante el desarrollo o la explotación
- Defectos o incidencias
- Estabilidad

A menudo se proponen medidas de atributos externos en función de atributos internos. Ejemplo: la efectividad del mantenimiento del código puede medirse en función de la proporción entre el número de errores descubiertos y el número de errores corregidos.

Medidas Directas

- Medidas Directas del proceso:
 - Costo y esfuerzo.
- Medidas Directas del producto:
 - Líneas de Código (LDC) Producidas, velocidad de ejecución, tamaño de memoria, Defectos informados durante un período.

Medidas Indirectas:

- funcionalidad, calidad, complejidad, eficiencia, fiabilidad, facilidad de mantenimiento, facilidad de uso.
- Es necesario normalizar las medidas para crear métricas de software que permitan comparar diferentes proyectos.

Métricas de seguimiento y control de proyectos

Las métricas de seguimiento y de control de proyectos intentan cuantificar y predecir un valor de la calidad media con que el proyecto está avanzando.

Podemos encontrar algunas métricas que nos permitirán realizar el seguimiento y control, así como se muestra en la tabla 2.5:

Tabla 2.5 Métricas de seguimiento y control

Métrica	Características
Métrica de documentación	Esta métrica nos indica la proporción de la documentación aportada por el equipo de trabajo con respecto al tamaño del

	programa.
Métrica de densidad de fallos	Recoge datos sobre el número de defectos descubiertos durante cierta fase de pruebas.
Métrica de propagación de errores	Esta métrica recoge datos sobre el número de defectos que aparecen al corregir un error
Métrica de disponibilidad	Indica el porcentaje de tiempo que el sistema está funcionando correctamente.
Métrica de redundancia de funcionalidad	Esta métrica nos indica el porcentaje de código redundante que existe en el sistema.

La métrica de documentación podría ser calculada en base a la relación entre el número de documentos generados y el número de documentos que fueron modificados por no cumplir con los estándares de calidad.

Métricas de Estimación

Los puntos de función es una métrica de estimación que se basa no en las LDC sino en una métrica que cuantifica la funcionalidad que hay que entregar al usuario al construir una aplicación, ésta métrica se detallara en la siguiente lección.

¿Qué es una estimación?

Estimar consiste en determinar el valor de una variable desconocida a partir de otras conocidas, o de una pequeña cantidad de valores conocidos de esa misma variable.

Métricas de calidad

Son todas las métricas de software que definen de una u otra forma la calidad del software; tales como exactitud, estructuración o modularidad, pruebas, mantenimiento, reusabilidad, cohesión del módulo, acoplamiento del módulo, etc. Estas son los puntos críticos en el diseño, codificación, pruebas y mantenimiento.

Es difícil, desarrollar medidas directas de factores de calidad del software. Cada factor de calidad F_c se puede obtener como combinación de una o varias métricas:

$$F_c = c_1 * m_1 + c_2 * m_2 + \dots + c_n * m_n$$

Donde:

- c_i es el factor de ponderación de la métrica i , que dependerá de cada aplicación específica
- m_i métrica i (métrica de documentación, esfuerzo, % errores, etc)

2.3.4 LECCIÓN No. 4: Método de estimación de puntos de función

Uno de los métodos de estimación de proyectos de software muy utilizado es el de Puntos de Función, el cual exige tener establecidos previamente los requisitos funcionales del aplicativo, así como también el modelo Entidad Relación, los mismos que apoyarían a una estimación más real del proyecto.

El método de estimación por puntos de función definido inicialmente por Albrecht en IBM, 1979, se caracteriza por medir aspectos funcionales del dominio del problema.

Proceso del método de Puntos de Función

El proceso para obtener los puntos de función constituye en tres pasos fundamentales:

1. Obtención de los puntos de función sin ajustar
2. Definición del factor de ajuste
3. Obtención de los puntos de función ajustados

Obtención de los puntos de función sin ajustar

Parámetros Básicos

En este punto es necesario clasificar a la funcionalidad del aplicativo en cinco parámetros básicos, los cuales se agrupan en parámetros de datos y parámetros de transacciones, así como se muestra en la Tabla 2.6

Tabla 2.6: Parámetros Básicos

		Fichero Lógico Interno (Internal Logic File)
--	--	---

Parámetros de Datos	ILF	Grupos de datos lógicamente relacionados identificables por los usuarios o información de control mantenidos y utilizados dentro de los límites de la aplicación
	EIF	Fichero de interfaz externa (External Interface File) Grupos de datos relacionados lógicamente identificables por el usuario o información de control utilizada por la aplicación, pero mantenida por otro sistema. Son externos a la aplicación que está siendo contada.
Parámetros de Transacciones	EI	Entrada Externa (External Input) Datos o información de control que se introduce en la aplicación desde fuera de sus límites para añadir modificar o eliminar los datos en un fichero lógico interno.
	EO	Salida Externa (External Output) Datos o información de control que sale de los límites de la aplicación, constituyen reportes de datos derivados o calculados de la información contenida en los ILF y EIF.
	EQ	Consulta Externa (External Query) Datos no calculados que se obtienen por la combinación de una EI y de una EO, ningún ILF se modifica en un proceso de consulta.

Una vez comprendido en que consiste cada elemento, es importante lograr la habilidad de determinar en que elemento se incluye cada funcionalidad encontrada en el aplicativo.

ILF Fichero Lógico Interno (Internal Logic File)

Reglas de identificación:

- El ILF está mantenido dentro de las fronteras del proyecto.
- El ILF está modificado y mantenido por un proceso elemental de la aplicación.

- El ILF no puede ser identificado como un EIF.

Ejemplos:

- Ficheros Maestros (almacenes).
- Aplicaciones de seguridad de datos
- Datos de auditoría
- Mensajes help actualizados por la aplicación.
- Mensajes de error actualizados por la aplicación.
- Datos de Backup, si el usuario lo requiere.

EIF Fichero Lógico Externo (External Interface File)

Reglas de identificación:

- El grupo de datos es referenciado y es externo a la aplicación que ésta siendo contada.
- El grupo de datos no es mantenido por la aplicación que está siendo contada.
- El grupo de datos se ha contado como un ILF en otra aplicación.
- El grupo de datos identificado no ha sido contado como un ILF para la aplicación.

EI Entrada Externa (External Input)

Reglas de Identificación:

- Los datos se reciben desde fuera de los límites de la aplicación.
- Los datos mantienen un ILF a través de un proceso elemental de la aplicación.
- El proceso es la unidad más pequeña de actividad que es significativa para el negocio del usuario final.
- El proceso es autocontenido y deja a la aplicación que está siendo contada en un estado consistente.
- El proceso identificado debe verificar alguna de las dos reglas:
 - Su lógica de proceso es única respecto a otras entradas externas de la aplicación.
 - Los elementos de datos identificados son distintos a los de las otras Entradas Externas de la aplicación.

Ejemplo de entradas internas:

- Transacciones: Datos introducidos para mantener los ILF
- Pantallas de entrada
- Entradas por lotes

EO Salidas Externas (External Output)**Reglas de Identificación:**

- El proceso envía datos o información de control
- Los datos o información de control se envían a través de un proceso elemental de la aplicación
- El proceso es la unidad más pequeña de actividad que es significativa para el negocio del usuario final
- El proceso es autocontenido y deja a la aplicación en un estado consistente
- El proceso identificado debe verificar alguna de las dos reglas:
 - Su lógica de proceso es única respecto a otras salidas externas de la aplicación.
 - Los elementos de datos identificados son distintos a los de las otras Salidas Externas de la aplicación.

Ejemplo de salidas externas:

- Transferencias de datos a otras aplicaciones
- Los informes con datos calculados o no calculados
- Los gráficos
- Informes on-line

EQ Consultas Externas (External Query)**Reglas de Identificación:**

- Una petición entra dentro del límite de la aplicación
- Un resultado sale del límite de la aplicación
- Hay recuperación de datos
- Los datos recuperados no contienen datos derivados

- La petición de entrada y el resultado de salida juntos, hacen del proceso la unidad de actividad más pequeña que es significativa para el negocio del usuario final
- El proceso no actualiza ILFs
- El proceso identificado debe verificar alguna de las dos reglas:
 - La lógica de proceso sobre la entrada y la salida es única respecto a otras consultas externas de la aplicación.
 - Los elementos de datos que forman la entrada y la salida son distintos a los de las otras consultas externas de la aplicación.

Ejemplo de consultas externas:

- Búsqueda inmediata de datos, ayudas, tutoriales
- Un informe con datos no calculados
- Menús con consultas
- Pantallas de conexión

Valoración de la complejidad

Una vez definido cada parámetro de la funcionalidad del aplicativo es necesario valorar su complejidad.

En los parámetros orientados a datos la complejidad se asigna en función de:

- 1. DET Tipo de Elemento de datos:** es un campo único y no recursivo de los ILF o EIF.
- 2. RET Tipo de Elemento de Registro:** es un subgrupo de datos elementales de los ILF o EIF.

En los parámetros orientados a la transacción la complejidad se asigna en función de:

- 1. DET Tipo de Elemento de datos:** es un campo único y no recursivo de los ILF o EIF.
- 2. FTR Fichero Referenciado:** es un ILF leído o mantenido, o un EIF leído por un parámetro de transacción.

Valoración de la complejidad de un ILF o EIF

Para determinar la complejidad de un ILF o EIF es necesario definir cuantos DET y RET utiliza. Por ejemplo, la aplicación tiene una tabla Cliente, con los datos: nombre, dirección, teléfono. La cantidad de DET= 3, cantidad RET = 1. Utilizando la tabla 2.7 se define que la complejidad de ésta ILF es baja.

Tabla 2.7: Valores de ILF o EIF

	DET		
RET	1 a 19	20 a 50	51 o más
1	Baja	Baja	Media
2 a 5	Baja	Media	Alta
6 o más	Media	Alta	Alta

Valoración de la complejidad de un EI o EO

Para obtener la complejidad de un EI o EO es necesario definir cuantos DET y FTR utiliza. Por ejemplo, la aplicación para añadir un libro, tiene que ingresar datos del Libro: Nombre, año publicación y Editorial; y del Autor: Nombre, Dirección. La cantidad de DET= 5, cantidad FTR = 2 (libro y autor). Utilizando la tabla 2.8 se define que la complejidad de ésta EI es media.

Tabla 2.8: Valores de EI o EO

	DET		
FTR	1 a 4	5 a 15	16 o más
0 a 1	Baja	Baja	Media
2	Baja	Media	Alta
3 o más	Media	Alta	Alta

Valoración de la complejidad de un EQ

Para determinar la complejidad de una EQ es necesario recordar que éste parámetro consta de una entrada externa EI y de una salida externa EO, por lo tanto es preciso valorar la complejidad por separado de la EI y de la

EO basándose en la Tabla 2.8 seleccionando a la de mayor complejidad. Por ejemplo: tengo una consulta cuya EI es complejidad baja y su EO es complejidad media, la EQ finalmente tendrá complejidad media.

Obtención de los puntos de función sin ajustar

Con el análisis explicado en los puntos anteriores se logrará determinar la cantidad de parámetros básicos clasificados por su complejidad en la tabla 2.9:

Tabla 2.9: Puntos de función sin ajustar

Parámetros	Bajo	Medio	Alto
ILF	5	2	
EIF			2
EI	6		1
EO	4	2	
EQ		4	

De acuerdo a la tabla ejemplo, la aplicación tendría 5 ILF con complejidad Baja, 4 EQ complejidad Media, etc.

La técnica de estimación por puntos se función establece pesos específicos para cada parámetro básico y su complejidad, detallados en tabla 2.10:

Tabla 2.10: Pesos específicos para cada parámetro básico

Parámetros	Bajo	Medio	Alto
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

Siguiendo la secuencia de cálculo, se procederá a multiplicar cada valor obtenido de los parámetros básicos por el peso respectivo mostrada en la tabla 2.11. Por ejemplo basándonos en la tabla 4.6 los puntos de función sin ajustar sería 1004.

Tabla 2.11: Cálculo de puntos de función

Parámetros	Bajo	Medio	Alto	Total
ILF	5 x 7	2 x 10		700
EIF			2 x 10	20
EI	6 x 3		1 x 6	108
EO	4 x 4	2 x 5		160
EQ		4 x 4		16
				1004

Definición del Factor de Ajuste

Determinación del valor del grado de influencia GDI

Para obtener el grado de influencia GDI debe efectuarse el análisis de las características generales del sistema, las cuales son:

Tabla 2.12: Grados de influencia

Características generales del sistema	Grados de influencia 0-5
1. Comunicación de datos	
2. Sistema distribuido	
3. Rendimiento	
4. Configuraciones fuertemente utilizadas	
5. Frecuencia de transacciones	
6. Teleprocesos	
7. Diseño para la eficiencia del usuario final	
8. Teleprocesos de Actualización	
9. Procesos complejos	
10. Utilización en otros sistemas (reutilización)	
11. Facilidad de instalación	
12. Facilidad de operación	
13. Múltiple instalación	

14. Facilidad de cambio	
Grados de influencia Total (GDI)	

A cada característica se le debe valorar en una escala de 0 a 5, como lo muestra la tabla 2.13:

Tabla 2.13: Escala GDI

0	Inexistente o sin influencia
1	Influencia secundaria
2	Influencia restringida
3	Influencia media
4	Influencia importante
5	Influencia intensiva e importante

A continuación se procederá a explicar cada característica general del sistema y las condiciones que encaminaran a determinar su valoración.

1. Comunicación de datos

Los datos e información de control utilizados en el sistema están transmitidos a través de herramientas de telecomunicación.

Tabla 2.14: Evaluación de comunicación de datos

Evaluación	Descripción
0	La aplicación es por lotes o utilizando un ordenador personal
1	Impresión o una entrada de datos remotas
2	La aplicación es por lotes pero son remotas las entradas de datos o la impresión
3	Entrada on-line a un proceso por lotes o sistema de consultas
4	Más de un ordenador front-end pero la aplicación soporta un solo tipo de protocolo de comunicaciones
5	Más de un ordenador front-end, pero la aplicación soporta mas de un tipo de protocolo de comunicaciones

2. Funciones Distribuida

Los componentes de la aplicación están distribuidos en dos o más procesadores diferentes.

Tabla 2.15: Evaluación de funciones distribuidas

Evaluación	Descripción
0	La aplicación no ayuda a la transferencia de datos
1	La aplicación prepara datos para el usuario final de otro procesador
2-3	Los datos se preparan para transferencia, se transfieren y se procesan en otro componente del sistema
4	Igual que 2-3, pero con realimentación al sistema inicial
5	Las funciones de procesamiento se realizan dinámicamente en el componente más apropiado del sistema

3. Rendimiento

Importancia de respuesta dentro de todo el sistema.

Tabla 2.16: Evaluación del rendimiento

Evaluación	Descripción
0-3	Análisis y diseño de las consideraciones del rendimiento son estándar. No se requieren requerimientos especiales por parte del usuario
4	Los requisitos de rendimiento por parte de los usuarios son suficientemente estrictos como para requerir un análisis de rendimiento en la fase de diseño
5	Además, hay que utilizar herramientas para análisis de rendimiento durante el diseño, desarrollo y/o fase de implantación para verificar los requisitos de rendimiento

4. Configuraciones fuertemente utilizadas

Referente a la importancia del entorno.

Tabla 2.17: Evaluación de configuraciones fuertemente utilizadas

Evaluación	Descripción
0-3	La aplicación corre en una máquina estándar sin restricciones de operación

4	Restricciones de operación requieren características específicas de la aplicación en el procesador central
5	Además de las características del punto 4 existen limitaciones en los componentes distribuidos del sistema

5. Frecuencia de transacciones

Si la frecuencia de transacciones es alta, influye sobre el diseño, desarrollo, instalación y soporte de la aplicación.

Tabla 2.18: Evaluación de frecuencia de transacciones

Evaluación	Descripción
0	No existe una definición del periodo punta de transacciones
1	Se conoce el periodo punta (mensual, trimestral, estacional, anual)
2	Se conoce el periodo semanal
3	Se conoce el periodo punta diario
4	En la fase de diseño se incluyen tareas de análisis de rendimiento para verificar las altas tasas de transacciones
5	Además se utilizan herramientas de análisis del rendimiento.

6. Entradas de datos on-line

Tabla 2.19: Evaluación de datos on-line

Evaluación	Descripción
0-2	Hasta el 15% de las transacciones tienen entrada interactiva
3-4	15% al 30% tienen entrada interactiva
5	30% al 50% tienen entrada interactiva

7. Eficiencia del usuario final

Las funciones on-line proporcionadas ponen énfasis en un diseño que incremente la eficiencia del usuario final. Estas funciones pueden ser:

- Ayudas a la navegación (teclas de función...)
- Menús
- Pantallas de ayuda on-line

- Movimiento automático del cursor
- Impresión remota (por transacciones on-line)
- Teclas de función preasignadas
- Sumisión de trabajos por lotes a través de teleprocesos
- Selección de datos con el cursor
- Fuerte uso de facilidades de vídeo
- Documentación usuario en formato papel y on-line
- Ventanas
- Soporte bilingüe y multilingüe

Las descripciones para determinar el grado de influencia son:

Tabla 2.20: Evaluación de eficiencia del usuario final

Evaluación	Descripción
0	Ninguno de los puntos anteriores
1	1 a 3 de los puntos anteriores
2	4 a 5 de los puntos anteriores
3	6 o más, pero no existen requisitos del usuario respecto a la eficiencia.
4	6 o más pero están definidos los requisitos de eficiencia del usuario que obligan a diseñar tareas que tienen en cuenta factores humanos; por ejemplo, minimizar el número de tecleos, uso de mascarar, etc.
5	6 o más, y hay requisitos del usuario sobre eficiencia que obligan a utilizar herramientas especiales y procesos para demostrar que los objetivos se han alcanzado.

8. Actualizaciones on-line

La aplicación proporciona actualizaciones on-line de los ficheros lógicos internos.

Tabla 2.21: Evaluación de actualizaciones on-line

Evaluación	Descripción
------------	-------------

0	Ninguno
1-2	Actualización on-line de los ficheros de control. El volumen de actualización es bajo y la recuperación fácil.
3	Actualización on-line de la mayoría de los ficheros internos lógicos
4	Además es esencial la protección contra la pérdida de datos
5	Además se considera el coste de recuperación de volúmenes elevados.

9. Procesos complejos

La complejidad de los procesos constituye una característica de la aplicación si en algunos de los componentes están presentes:

- Controles especiales (proceso de auditoría especial) o Procesos de seguridad específicos.
- Procesos lógicos complejos
- Procesos matemáticos complejos
- Procesos de excepción (Transacción de tipo ATM incompleta debida a interrupciones).
- Procesos complejos de manejo de múltiple entrada/Salida (Multimedia)

Tabla 2.22: Evaluación de procesos complejos

Evaluación	Descripción
0	Ningún componente anterior
1-3	Existe algún componente presentado anteriormente.
4	Existe 4 de los componentes presentados anteriormente.
5	Existe 5 de los componentes presentados anteriormente.

10. Utilización en otros sistemas (reutilización)

La aplicación y el código han sido diseñados específicamente, desarrollados y soportados para ser utilizados en otras aplicaciones.

Tabla 2.23: Evaluación de utilización en otros sistemas

Evaluación	Descripción
------------	-------------

0	El código no es reusable
1	Se utiliza código ya existente en la aplicación
2	Menos de 10 % de la aplicación es de código reusable (módulos...)
3	Más de 10 % de la aplicación es de código reusable (módulos...)
4	Aplicación específicamente diseñada para su reuso con una adaptación del código fuente.
5	Aplicación específicamente diseñada para su reuso con una adaptación por parametrización sin cambio en el código fuente.

11. Facilidad de instalación

Las facilidades de conversión o instalación incrementa las dificultades del desarrollo de la aplicación.

Tabla 2.24: Evaluación de facilidad de instalación

Evaluación	Descripción
0-1	No se requieren por parte del usuario facilidades especiales de conversión e instalación
2-3	Los requisitos de conversión e instalación fueron definidos por el usuario y las guías para la conversión e instalación fueron proporcionadas y probadas.
4-5	Además se proporcionaron y probaron herramientas de conversión e instalación

12. Facilidad de operación

Las facilidades de operación requieren un plan establecido. Su propósito es proporcionar unos procesos de arranque, backup, etc, minimizando las intervenciones manuales.

Tabla 2.25: Evaluación de facilidad de operación

Evaluación	Descripción
0	No se definieron por parte del usuario necesidades especiales de operación.
1-4	Seleccionar, valorando como uno, cada una de las siguientes solicitudes realizadas a la aplicación: <ul style="list-style-type: none"> ○ Procesos eficientes de arranque, respaldo y recuperación pero con

	<p>intervención del operador (contar como 2)</p> <ul style="list-style-type: none"> ○ La aplicación minimiza la necesidad de montaje de cintas ○ La aplicación minimiza la necesidad de manejo de papel
5	La aplicación debe diseñarse sin intervención de operadores; es decir el ordenador no debe intervenir más que para arrancar y parar la aplicación. Uno de los elementos de la aplicación es la recuperación automática de errores.

13. Instalación en distintos lugares

La aplicación se diseñará y desarrollará para ser instalada y mantenida en distintos lugares por distintas organizaciones.

Tabla 2.26: Evaluación de instalación en distintos lugares

Evaluación	Descripción
0	El usuario no requiere la consideración de más de un puesto
1-2	De uno a cuatro puestos
3	Cinco o más puestos
4	Se proporciona documentación y plan de apoyo para soportar la aplicación en varios lugares
5	Los puestos están en países diferentes

14. Facilidad de cambio

La aplicación fue diseñada, desarrollada, y mantenida para facilitar los cambios.

Tabla 2.27: Evaluación de facilidad de cambio

Evaluación	Descripción
0	No hay requerimientos especiales del usuario para minimizar o facilitar el cambio
1-3	Se proporciona capacidad de consulta flexible
4-5	Datos importantes de control se mantienen en tablas que son actualizadas por el usuario a través de procesos on-line interactivos

Cada una de las características tendrá una única valoración. El GDI se obtendrá de la suma de los valores de cada una de ellas. Por ejemplo se a definido la siguiente valoración para cada característica obteniendo un resultado de GDI = 28:

Tabla 2.28: GDI

Características generales del sistema	Grados de influencia 0-5
1. Comunicación de datos	2
2. Sistema distribuido	4
3. Rendimiento	1
4. Configuraciones fuertemente utilizadas	0
5. Frecuencia de transacciones	3
6. Teleprocesos	2
7. Diseño para la eficiencia del usuario final	5
8. Teleprocesos de Actualización	4
9. Procesos complejos	0
10. Utilización en otros sistemas (reutilización)	2
11. Facilidad de instalación	1
12. Facilidad de operación	3
13. Múltiple instalación	1
14. Facilidad de cambio	0
Grados de influencia Total (GDI)	28

Obtención de los puntos de Función Ajustados

Una vez calculado el grado de influencia GDI se puede llegar a obtener el factor de ajuste, de acuerdo a la siguiente fórmula:

$$\mathbf{FA = 0.65 + (0.01 * GDI)}$$

FA: Factor de Ajuste (AF Adjustment Factor)

GDI: Grado de Influencia (TDI Total Degree of Influence)

Continuando con el ejemplo planteado en los ítems anteriores se obtendría el factor de ajuste:

$$\mathbf{FA = 0.65 + (0.01 * 28)}$$

$$\mathbf{FA = 0.93}$$

El valor final de los puntos de función ajustados se obtendrá atrás de la siguiente fórmula:

$$PF = FA * PFSA$$

En base al ejemplo, los puntos de función ajustados sería:

$$PF = 0.93 * 1004$$

$$PF = 933.72$$

Dentro de las ventajas de calcular puntos de función existe la posibilidad de determinar los siguientes aspectos.



Figura 2.21: Ventajas del cálculo de Puntos de Función

- ✓ Productividad: indica el número de puntos de función que puede desarrollar en un mes una persona.
- ✓ Calidad: indica el número de errores que supuestamente se cometerán por punto de función.
- ✓ Costo: indica los dólares que costará a la empresa el desarrollo de un punto de función.
- ✓ Documentación: indica el número de páginas de documentación que se generará por punto de función.
- ✓ Líneas de código: indica el número de líneas de un determinado lenguaje de programación que se escribirán por punto de función.

Los parámetros anteriormente señalados nos proporcionará datos importantes dentro del desarrollo del proyecto, a continuación con un ejemplo de una empresa se mostrará la relevancia de los resultados, para lo cual se considera los siguientes datos:

Tabla 2.29 Datos del ejercicio

Puntos de Función	61
Esfuerzo	20 PF/persona-mes
Cantidad de errores	7 errores/PF
Costo	\$ 1000 /PF
Cantidad de páginas de documentación	0.5 páginas de especificación de requisitos / PF
Número de líneas de código	105 instrucciones COBOL / PF

Productividad = PF / Esfuerzo(persona – mes)

Reemplazando los datos se obtiene

$$\frac{61 \text{ PF}}{20 \text{ PF/persona- mes}} = 3 \text{ persona-mes}$$

El número de errores viene dado por No. Errores (defectos) x PF

$$7 \text{ errores} \times 61 \text{ PF} = 427 \frac{\text{errores de línea de código}}{\text{PF}}$$

Costo de la aplicación es igual dólares x PF

$$\frac{\$ 1000 \times 61 \text{ PF}}{\text{PF}} = \$ 61000$$

La documentación es igual a páginas de documentación x PF

$$0.5 \text{ pág.} \times 61 \text{ PF} = 30.5 \frac{\text{pág.}}{\text{PF}} \approx 31 \text{ pág.}$$

El número de líneas de código es calculado mediante el # instrucciones x PF

$$\frac{105 \text{ inst} \times 61}{\text{PF}} = 6405 \text{ instrucciones}$$

Dentro de las ventajas del uso de puntos de función están:

- Ayudar a entender el crecimiento de proyectos.
- Calculo del costo real del software, la programación y el esfuerzo.
- Ayuda a entender los costos de mantenimiento.
- Los administradores de contratos pueden usar su conocimiento en puntos función para construir y manejar proyectos basados en el precio por punto función y también en la comparación de los precios de los vendedores.
- El uso de puntos función permite desarrollar un estándar de establecimiento de métricas.

Algunas de las desventajas de la utilización de puntos de función es que su apreciación está basada en historiales de proyectos. Segundo para el cálculo de los puntos de función los requerimientos funcionales deben estar bien identificados para que los resultados se acerquen a la realidad de lo que el planificador del proyecto debe considerar en la estimación.

2.3.5 LECCIÓN No. 5: Relación entre los puntos de función y COCOMO

El tamaño de una aplicación se mide en unidades de líneas de código fuente (KLOC). En COCOMO puede estimarse también a partir de los puntos de función sin ajustar convirtiendo a LOC y luego dividiendo por 1000.

Para convertir las líneas de código fuente en el lenguaje de programación nos basaremos en la siguiente tabla:

Tabla 2.30: Relación entre Líneas de Código y Puntos de Función

LOC por punto de función			
Lenguaje	LOC/PF	Lenguaje	LOC/PF
Ensamblador	320	Basic ANSI	64
Macroensamblador	213	Java	53
C	150	Visual C++	34
Fortran	106	Foxpro 2.5	34
Cobol	106	Visual Basic	32

Pascal	91	Delphi	29
Cobol ANSI 85	91	C++	29
Basic	91	Visual Cobol	20
RPG	80	Clipper	19
PL/1	80	Power Builder	16
Ada	71	Hoja de Cálculo	6

Finalmente para obtener KLOC se debe hacer la conversión, mediante la operación LOC/1000, así por ejemplo; si se obtiene como resultado 165 UNFP (Puntos de Función sin Ajustar) y el proyecto va a desarrollarse en el lenguaje de programación C++,

$$165 \text{ UNFP} \times 29 = 4785 \text{ SLOC (Líneas de Código Fuente)}$$

Haciendo la conversión mencionada anteriormente:

$$4785/1000 = 4.785 \text{ KLOC (Miles de Líneas de Código Fuente).}$$

COCOMO 81

Inventado por el profesor Barry Boehm. Se basa en el estudio de 63 proyectos software que cubren varios lenguajes y áreas de programación.

COCOMO distingue:

Modelos de Estimación

Tabla 2.31: Tipos de modelos de estimación

Modelo básico:	Productos pequeños medios
	Personal de la empresa
	Generalmente modo orgánico
Modelo intermedio	Multiplicadores de esfuerzo

Modelo detallado	Grandes proyectos
	Multiplicadores de esfuerzo por fases
	Jerarquización: Nivel de sistema, nivel de subsistema, nivel de módulos.

Modelo Básico

Cálculo del Esfuerzo

La siguiente fórmula relaciona esfuerzo al tamaño del software

$$E = aKLOC^b$$

a y b son constantes que dependen del tipo de proyecto que se ejecuta

KLOC son líneas de código en miles

Lo primero que necesitamos saber es el tipo de proyecto del que estamos hablando, hay tres tipos diferentes:

Tabla 2.32: Tipos de proyecto

Modo orgánico:	Desarrollado por la propia empresa
	Amplia experiencia del personal en dicho tipo de proyectos
	Entorno de desarrollo estable
	Buen entendimiento
	No hay restricciones de tiempo para acabar antes
Modo semi-acoplado:	Trabajos pequeños, 50 KLOC máximo
	Personal con experiencia intermedia
	Algunos tienen experiencia y otros no

	Algunas interfaces muy rigurosas otras flexibles
	Tamaño máximo 300 KLSI
Modo empotrado:	Limitaciones estrictas
	Entorno hardware y software muy restringido
	Alto coste para modificaciones
	Proyecto sobre áreas poco conocidas
	No hay limitación en el tamaño

El valor de cada constante según su modo:

Tabla 2.33: Constantes según su modo

MODO	a	b	c	d
Orgánico	2.40	1.05	2.50	0.38
Semi-acoplado	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.32

Reemplazando la fórmula del Esfuerzo relacionado al tamaño del software se obtienen los siguientes resultados.

Tabla 2.34: Resultados

KLOC	Orgánico	Semi-acoplado	Empotrado
	$E = 2.4KLOC^{1.05}$	$E = 3KLOC^{1.12}$	$E = 3.6KLOC^{1.20}$
1	2.4	3.0	3.6
10	26.9	39.6	57.1
50	145.9	239.4	392.9
100	302.1	521.3	904.2
1000	3390	6872.0	1433.0

Cálculo del tiempo de desarrollo del proyecto

Tomando la constante c indicada en la tabla 5.3, COCOMO determina el tiempo de desarrollo del proyecto (TDEV) (expresado en meses), usando el esfuerzo previamente calculado. Su fórmula es:

$$\text{TDEV} = c\text{Esfuerzo}^d$$

Reemplazando los valores obtendríamos lo siguiente:

$$\text{Orgánico TDEV} = 2.5 * \text{Esfuerzo}^{0.38}$$

$$\text{Semi-acoplado TDEV} = 2.5 * \text{Esfuerzo}^{0.35}$$

$$\text{Empotrado TDEV} = 2.5 * \text{Esfuerzo}^{0.32}$$

Cálculo de las personas necesarias para realizar el proyecto

Una vez determinado el esfuerzo y el tiempo en el que se llevará a cabo el proyecto, podemos calcular el número de personas necesarias mediante el reemplazo de los siguientes valores calculados anteriormente, así como lo muestra la siguiente fórmula:

$$P_E = \frac{\text{Esfuerzo}}{\text{TDEV}}$$

Cálculo del costo total del Proyecto

Es muy importante dentro del desarrollo de software, determinar el costo total del proyecto para eso se debe conocer el número de personas necesarias en el proyecto y multiplicar por el salario medio de los programadores y analistas.

$$\text{CostoM} = P_E * \text{Salario medio entre los programadores y analistas}$$

Cálculo

COCOMO se usa también para estimar el esfuerzo requerido para el mantenimiento de software.

$$\text{Esfuerzo}_{\text{mantenimiento}} = \text{ACT} * \text{Esfuerzo}$$

Donde ACT es la fracción de instrucciones modificadas por año; eso es igual $(\# \text{ de instrucciones añadidas} + \# \text{ de instrucciones cambiadas}) / \# \text{ instrucciones totales}$.

Modelo Intermedio

Dentro de las variaciones, éste modelo añade quince modificadores opcionales para tener en cuenta el entorno de trabajo, incrementando la precisión de la estimación. El resultado de este ajuste, al resultado de la fórmula general se lo multiplica por los atributos que se deciden utilizar.

Los valores de las constantes a reemplazar en la fórmula son:

Tabla 2.35: Constantes según su Modo

MODO	a	b
Orgánico	3.20	1.05
Semi-acoplado	3.00	1.12
Empotrado	2.80	1.20

Atributos

Cada uno de los atributos es cuantificado para un entorno de proyecto. La escala consiste de los siguientes puntos.

- VL (very low) (muy bajo)
- LO (low) (bajo)
- NM (nominal)
- HI (high) (alto)
- VH (very high) (muy alto)
- XH (extra high) (extra alto)

La lista de atributos está compuesta de varias características del software e incluye:

- producto,
- ordenador,
- personal,
- y atributos del proyecto

Atributos del producto

- Requisitos de fiabilidad (**RELY**)
Expresa el efecto de las fallas del software. Desde una simple inconveniencia (VL), hasta la pérdida de vida (VH). El valor nominal (NM) denota moderadas pérdidas que se pueden recuperar.
- Volumen de datos manipulados (**DATA**)
La evaluación más baja es para pequeñas bases de datos.
- Complejidad del producto (**CPLX**)
Expresa la complejidad de la programación. Desde programación batch (VL), hasta programación de tiempo real con uso de múltiples recursos (XH).

Atributos del ordenador

- Restricciones de tiempo de ejecución (**TIME**)
- Restricciones de tamaño de memoria (**STOR**)
Identifican el porcentaje de recursos (tiempo, memoria) usados por el sistema. NM es asignado cuando menos del 50% es usado. 95% es indicado por XH.
- Inestabilidad de la máquina virtual (**VIRT**)
Indica la frecuencia de cambios hechos al hardware, sistema operativo, y ambiente del software. Cambios más frecuentes y significativos son indicados con una evaluación más alta.
- Tiempo de respuesta experimentado por el equipo que desarrolla (**TURN**)
Tiempo que se demora un trabajo, desde que fue enviado hasta que el output se recibe. LO indica un ambiente altamente interactivo; VH cuantifica la situación cuando el tiempo de respuesta es > 12 horas

Atributos del personal

- Capacidad de los analistas (**ACAP**)
- Capacidad de los programadores (**PCAP**)

Describe la habilidad del equipo de desarrollo. Más habilidad se indica con una evaluación más alta.

- Experiencia en Aplicaciones (**AEXP**)
- Experiencia en el lenguaje de programación (**LEXP**)
- Experiencia en la máquina virtual (**VEXP**)

Cuantifica la experiencia que el equipo de desarrollo tiene en estas áreas. Más experiencia, más alta la evaluación.

Atributos del proyecto

- Prácticas modernas en programación (**MODP**)
- Uso de herramientas para el desarrollo de software (**TOOL**)

Mide el nivel de sofisticación de herramientas para desarrollo de software y el grado de integración entre las herramientas usadas. Una evaluación más alta significa niveles altos en ambos aspectos.

- Limitaciones en la planificación (**SCED**)

Una programación reducida (HI o VH). Una programación expandida (LO o VL) en comparación con una programación nominal (NM).

Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula. Por ejemplo, si para un proyecto el atributo DATA es calificado como muy alto (VH) el resultado de la fórmula debe ser multiplicado por 1000.

El valor de cada atributo dependiendo de su clasificación es:

Tabla 2.36: Atributos COCOMO 81

	VL	LO	NM	HI	VH	XH
RELY	0.75	0.88	1	1.15	1.4	
DATA		0.94	1	1.08	1.16	
CPLX	0.70	0.85	1	1.15	1.3	1.65
TIME			1	1.11	1.3	1.66

STOR			1	1.06	1.21	1.56
VIRT		0.87	1	1.15	1.3	
TURN		0.87	1	1.07	1.15	
ACAP	1.46	1.19	1	0.86	0.71	
AEXP	1.29	1.13	1	0.91	0.82	
PCAP	1.42	1.17	1	0.86	0.70	
LEXP	1.14	1.07	1	0.95		
VEXP	1.21	1.10	1	0.90		
MODP	1.24	1.10	1	0.91	0.82	
TOOL	1.24	1.10	1	0.91	0.83	
SCED	1.23	1.08	1	1.04	1.10	

Cada atributo es evaluado y estos resultados parciales son multiplicados dando como resultado un multiplicador (P)

Cálculo del Esfuerzo

La fórmula de esfuerzo es expresada de la siguiente manera:

$$E = \text{Esfuerzo}_{\text{nom}} * P$$

Esfuerzo_{nom} se calcula de la siguiente manera dependiendo del tipo de software:

Orgánico Esfuerzo_{nom} = 3.20 * KLOC^{1.05}

Semi-acoplado Esfuerzo_{nom} = 3.00 * KLOC^{1.12}

Empotrado Esfuerzo_{nom} = 2.80 * KLOC^{1.20}

Estas formulas reflejan esencialmente las mismas relaciones que en el modelo básico con cambios solo en el parámetro a.

Si algún atributo es evaluado como nominal, este no tiene efecto sobre el valor de P, y por lo tanto no cambia el valor de esfuerzo.

Modelo Detallado

Siendo éste un modelo para grandes proyectos, no será estudiado en ésta materia.

COCOMO II

La evolución a COCOMO II fue para desarrollar un modelo de estimación de tiempo y costo de software de acuerdo a los ciclos de vida utilizados en los 90 y en la primera década del 2000. Así como también proporcionar un marco analítico cuantitativo, un conjunto de herramientas y técnicas para la evaluación de los efectos de mejora tecnológica del software.

COCOMO II surge de la diversidad del mercado actual y el futuro del desarrollo del software. Las aplicaciones desarrolladas por usuarios finales con el fin de obtener soluciones rápidas y flexibles, sistemas de gran escala con alto grado de integración sin datos en el mercado que se pueda tomar como base, soluciones a problemas de infraestructura de procesamiento distribuido, procesamiento de transacciones a soluciones middleware; son algunas de las tantas necesidades que son adaptadas por los tres modelos de COCOMO II.

Se han añadido drivers de costo: DOCU, RUSE, PVOL, PEXP, LTEXT, PCON y SITE

Se han eliminado los drivers de costo: VIRT, TURN, VEXP, LEXP Y MODP

COCOMO II está compuesto por tres modelos denominados: Composición de Aplicación, Diseño Inicial y Post-Arquitectura.

Tabla 2.37: Características de los modelos de COCOMO II

Composición de Aplicación	Se emplea en desarrollos de software que se construyen a partir de componentes pre-empaquetados.
Diseño Inicial	Es utilizado en las primeras etapas del desarrollo en las cuales se evalúa las alternativas de hardware y software de un proyecto. Se usa Puntos Función, para estimar tamaño y el uso de un número reducido de factores de costo.

Post-Arquitectura	<p>Se aplica en la etapa de desarrollo, después que se haya definido la arquitectura del sistema, y en la etapa de mantenimiento. Este modelo utiliza:</p> <ul style="list-style-type: none"> • Puntos Función o Líneas de Código Fuente para estimar el tamaño. • 17 atributos (factores de costo) • 5 factores que determinan un exponente, que incorpora al modelo el concepto de deseconomía y economía de escala.
--------------------------	---

Diseño Inicial y Post-Arquitectura

Estimación del Esfuerzo Nominal

Recordando la fórmula del cálculo del Esfuerzo

$$PM_{\text{NOMINAL}} = A * (\text{SIZE})^B$$

Donde:

A representa los efectos multiplicativos de esfuerzo en proyectos de tamaño incremental. Provisionalmente se le ha estimado un valor de 2.45

Variable SIZE viene dado por la siguiente fórmula

$$Size = size \times \left[1 + \frac{BRAK}{100} \right]$$

Este porcentaje de Rotura **BRAK** sirve para ajustar el tamaño eficaz del producto, el factor BRAK no se usa en el Modelo de Composición de Aplicaciones. Por ejemplo un proyecto que parte de 100.000 instrucciones (Size= 100.000) pero desecha el equivalente a 20.000 instrucciones tiene un valor de BRAK de 20. Esto debe usarse para ajustar el tamaño efectivo del proyecto a 120.000 instrucciones para una estimación COCOMO II. (Size = 100.000 x [1 + (20/100)])

Variable B (ahorro y gasto software de escala)

$$B = 1.01 + 0.01x \sum_{j=1}^5 SF_j$$

Este exponente considera los gastos y ahorros relativos de escala encontrados en proyectos de software de distinto tamaño.

B<1 Proyecto presenta ahorro de escala

B=1 Los ahorros de escala y los gastos están equilibrados

B>1 El proyecto presenta gasto de escala

El exponente B se obtiene mediante los drivers de escala. Cada driver de escala tiene un rango de niveles de valores desde VL hasta XH. Cada nivel de valores tiene un peso, SF, y el valor específico del peso se llama factor de escala. Un factor de escala de un proyecto SF_j se calcula sumando todos los factores.

Los factores de escala son:

- **PREC** (precedentes) toma en cuenta el grado de experiencia previa en relación al producto a desarrollar, tanto en aspectos organizacionales como en el conocimiento del software y hardware a utilizar.
- **FLEX** (Flexibilidad de desarrollo) considera el nivel de exigencia en el cumplimiento de los requerimientos preestablecidos, plazos de tiempos y especificaciones de interfase.
- **RESL** (Arquitectura /resolución de riesgos) Este factor involucra aspectos relacionados al conocimiento de los ítems de riesgo crítico y al modo de abordarlos dentro del proyecto.
- **TEAM** (cohesión del equipo) toma en cuenta las dificultades de sincronización entre los participantes del proyecto: usuarios, clientes, desarrolladores, encargados de mantenimiento, etc. Estas dificultades pueden surgir por diferencias culturales, dificultad en la conciliación de objetivos, falta de experiencia y familiaridad con el trabajo en equipo.

- **PMAT** (Madurez del proceso) se basa en el Modelo de CMM propuesto por el Software Engineering Institute, el resultado de la evaluación según CMM es:

Tabla 2.38: Evaluación CMM

Nivel CMM	PMAT
1 – Mitad inferior	Muy Bajo
1 – Mitad superior	Bajo
2	Nominal
3	Alto
4	Muy Alto
5	Extra Alto

Cada uno de los factores de escala anteriormente señalados muestran diversos criterios y niveles de cuantificación los cuales son:

Tabla 2.39: Criterios de factores de escala

Factores de Escala	VL	LO	NM	HI	VH	XH
PREC	Completa	Completa	Algo	Familiar	Muy Familiar	Absolutamente Familiar
FLEX	Riguroso	Ocasional	Algo de relajación	Generalmente conforme	Algo conforme	Objetivos generales
RESL	Poco (20%)	Algo (40%)	A menudo (60%)	Generalmente (75%)	Mayormente (90%)	Totalmente (100%)
TEAM	Interacción muy difícil	Algo de dificultad de interacción	Hay interacción operativa	Cooperativa	Altamente cooperativa	Interacción total
PMAT	Promedio de respuestas afirmativas en el cuestionario CMM					

Los valores que asume cada uno de estos factores en cada nivel se presenta en la tabla 2.40

Tabla 2.40: Factores de escala para el modelo de COCOMO II de Diseño Inicial

Factores de Escala	VL	LO	NM	HI	VH	XH
PREC	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	5.07	4.05	3.04	2.03	1.01	0.00
RESL	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	7.80	6.24	4.68	3.12	1.56	0.00

Por ejemplo, si tenemos factores de escala con valores Muy Bajo, entonces un proyecto 100 KLOC tendrá $SF_j = (6.20+5.07+7.07+5.48+7.80) = 31.62$, $B = 0.91 + (0.01 \times 31.62) = 1.2262$ y un esfuerzo relativo $E = (100)^{1.23} = 288$ PM.

Ajuste mediante drivers de Costo

Los drivers de costo se usan para capturar características del desarrollo del software que afectan el esfuerzo para completar el proyecto. Los drivers de costo tienen un nivel de medida que expresa el impacto del driver en el esfuerzo de desarrollo. Estos valores pueden ir desde XL hasta XH.

$$MM = A \times (\text{SIZE})^B \times \pi \text{EM}_i$$

Los EM (multiplicador de esfuerzo) se usa para ajustar el esfuerzo Meses-persona nominal. Hay 7 multiplicadores de esfuerzo para el Modelo de Diseño Inicial y 17 para el modelo de Post-Arquitectura.

Tabla 2.41: Drivers de Diseño Inicial y Post-Arquitectura

Drivers Diseño Inicial	Drivers Post-Arquitectura
RCPX	RELY, DATA, CPLX, DOCU
RUSE	RUSE
PDIF	TIME, STOR, PVOL
PERS	ACAP, PCARP, PCON
PREX	AEXP, PEXP, LTEX
FCIL	TOOL, SITE
SCED	SCED

Multiplicadores de Esfuerzo en el Modelo Post-Arquitectura

Los drivers de costo para el modelo Post-Arquitectura son:

Factores del producto

Se refieren a las restricciones y requerimientos sobre el producto a desarrollar.

- **RELY** (Confiabilidad requerida) Este factor mide la confiabilidad del producto de software a ser desarrollado.

Tabla 2.42: Calificación RELY

	VL	LO	NM	HI	VH
RELY	Inconvenientes imperceptibles	Bajo, y con pérdidas fácilmente recuperables	Moderado, con pérdidas de fácil recuperación	Altas pérdidas financieras	Riesgo para la vida humana

- **DATA** (Tamaño de la base de datos) El esfuerzo requerido para desarrollar un producto de software está relacionado con el tamaño de la base de datos asociada. Su medida es calculada mediante la siguiente fórmula:

$$D/P = \text{Tamaño de la BDD}(\text{bytes}) / \text{Tamaño del programa} (\text{SLOC})$$

DATA es valorada como Bajo si D/P es menor que 10 y muy alto si es mayor que 1000.

- **CPLX** (Complejidad del producto) se divide en cinco áreas: Operaciones de Control, Operaciones de Cálculo, Dependencia de Dispositivos, Manejo de Datos e Interfaces de Usuario cada uno de ellas es valorada desde muy bajo hasta muy alto.
- **RUSE** (Requerimientos de reusabilidad) Este factor considera el esfuerzo adicional necesario para construir componentes que puedan ser reusadas dentro de un mismo proyecto o en futuros desarrollos.

Tabla 2.43: Calificación RUSE

	VL	LO	NM	HI	VH	XH
RUSE		Nada	Por proyecto	Por programa	Por línea de producto	Por múltiples líneas de código

- **DOCU** (Documentación) evalúa los requerimientos de documentación a lo largo del ciclo de vida del proyecto.

Tabla 2.44: Calificación DOCU

	VL	LO	NM	HI	VH
DOCU	Muchas etapas del ciclo de vida sin documentación	Algunas	De acuerdo a las necesidades exactas de las etapas del ciclo de vida	Excesiva	Muy Excesiva

Factores de la plataforma

- **TIME** (Restricción del tiempo de ejecución) Este factor representa la disponibilidad de tiempo de ejecución que será usado por el sistema, versus los recursos disponibles.

Tabla 2.45: Calificación TIME

	VL	LO	NM	HI	VH	XH
TIME			≤ 50%	70%	85%	95%

- **STOR** (Restricción del almacenamiento principal) Este factor es una función que representa el grado de restricción del almacenamiento principal impuesto sobre un sistema de software.

Tabla 2.46: Calificación STOR

	VL	LO	NM	HI	VH	XH
STOR			≤ 50%	70%	85%	95%

- **PVOL** (Volatilidad de la plataforma) Expresa la velocidad de cambio del hardware y software usados como plataforma.

Tabla 2.47: Calificación PVOL

	LO	NM	HI	VH
PVOL	Cambios cada año y menor cada mes	Mayor: 6 meses Menor: 2 semanas	Mayor: 2 meses Menor: 1 semana	Mayor : 2 semanas Menor: 2 días

Factores del personal

- **ACAP** (Capacidad de los analistas) y **PCAP** (Capacidad de los programadores) consideran la capacidad de análisis y diseño, eficiencia, habilidad para comunicarse y trabajar en equipo.

Tabla 2.48: Calificación ACAP y PCAP

	VL	LO	NM	HI	VH	XH
ACAP y PCAP	15%	35%	55%	75%	90%	

- **PCON** (Continuidad del Personal) expresa el porcentaje de rotación anual del personal afectado al proyecto.

Tabla 2.49: Calificación PCON

	VL	LO	NM	HI	VH	XH
PCON	48% al año	24% al año	12% al año	6% al año	3% al año	

- **AEXP** (Experiencia en aplicaciones) contempla el nivel de experiencia del grupo de desarrollo.
- **PEXP** (Experiencia en la plataforma) refleja la experiencia en el uso de herramientas de software y hardware utilizado como plataforma.
- **LTEX** (Experiencia en el Lenguaje y herramientas de desarrollo) Este factor mide el nivel de experiencia del equipo en el uso del lenguaje y herramientas a ser empleados.

Los 3 anteriores factores de personal podrían asumir estos valores:

Tabla 2.50: Calificación AEXP, PEXP y LTEX

	VL	LO	NM	HI	VH	XH
AEXP						
PEXP	2 meses	6 meses	1 año	3 años	6 años	
LTEX						

Factores del proyecto

- **TOOL** (Uso de herramientas de software) contempla el uso de herramientas, desde la edición hasta el manejo de todo el ciclo de vida.

Tabla 2.51: Calificación TOOL

	VL	LO	NM	HI	VH
TOOL	Editar código, depuración	CASE simple y de poca integración	Herramientas de ciclo de vida básicas	Potentes herramientas a ser usadas en todo el ciclo de vida con integración moderada	Herramientas potentes y proactivas, integradas con el proceso, métodos y reusabilidad

- **SITE** (Desarrollo en múltiples aplicaciones) Involucra la ubicación física y el soporte de comunicaciones.

Tabla 2.52: Calificación SITE

	VL	LO	NM	HI	VH	XH
SITE	Algo de teléfono y mail	Fax y teléfonos individuales	Banda estrecha email	Comunicación electrónica de banda ancha	Comunicación electrónica de banda ancha videoconferencia ocasional	Multimedia interactiva

- **SCED** (Requerimientos de calendario de desarrollo) refleja las restricciones impuestas al grupo de desarrollo sobre la agenda estimada del proyecto.

Tabla 2.53: Calificación SCED

	VL	LO	NM	HI	VH	XH
SCED	75% del nominal	85%	100%	130%	160%	

Los valores que asume cada factor de escala son:

Tabla 2.54: Drivers de Costo para el Modelo Post-Arquitectura

Factores de Escala	VL	LO	NM	HI	VH	XH
RELY	0.82	0.92	1.00	1.10	1.26	
DATA		0.90	1.00	1.14	1.28	
CPLX	0.73	0.87	1.00	1.17	1.34	1.74
RUSE		0.95	1.00	1.07	1.15	1.24
DOCU	0.81	0.91	1.00	1.11	1.23	
TIME			1.00	1.11	1.29	1.63
STOR			1.00	1.05	1.17	1.46
PVOL		0.87	1.00	1.15	1.30	
ACAP	1.42	1.19	1.00	0.85	0.71	
PCAP	1.34	1.15	1.00	0.88	0.76	
PCON	1.29	1.12	1.00	0.90	0.81	
AEXP	1.22	1.10	1.00	0.88	0.81	
PEXP	1.19	1.09	1.00	0.91	0.85	
LTEX	1.20	1.09	1.00	0.91	0.84	
TOOL	1.17	1.09	1.00	0.90	0.78	
SITE	1.22	1.09	1.00	0.93	0.86	0.80
SCED	1.43	1.14	1.00	1.00	1.00	

Por ejemplo, si al analizar los drivers de costo de nuestro proyecto obtenemos los siguientes resultados:

Tabla 2.55: Datos del ejemplo

Factores de Escala	VL	LO	NM	HI	VH	XH

RELY			X			
DATA			X			
CPLX		X				
RUSE			X			
DOCU					X	
TIME			X			
STOR			X			
PVOL			X			
ACAP				X		
PCAP			X			
PCON		X				
AEXP			X			
PEXP			X			
LTEX				X		
TOOL				X		
SITE			X			
SCED			X			

Aplicando los valores de la tabla 2.54 se obtiene:

$$\pi \mathbf{EM}_i = 1 \times 1 \times 0.87 \times 1 \times 1.23 \times 1 \times 1 \times 1 \times 0.85 \times 1 \times 1.12 \times 1 \times 1 \times 0.91 \times 0.90 \times 1 \times 1$$

$$= 0.834$$

Multiplicadores de Esfuerzo en el Modelo de Diseño Inicial

- **PERS** (Capacidad del personal) está dado por la suma o la combinación porcentual de los multiplicadores ACAP, PCAP y PCON.

Tabla 2.56: Calificación PERS

	XL	VL	LO	NM	HI	VH	XH
Suma de ACAP, PCAP, PCON	3,4	5,6	7,8	9	10,11	12,13	14,15
Combinación de ACAP y PCAP	20%	39%	45%	55%	65%	75%	85%
Rotación anual del personal	45%	30%	20%	12%	9%	5%	4%

- **RCPX** (Complejidad del Producto) combina los multiplicadores RELY, DATA, CPLX y DOCU.

Tabla 2.57: Calificación RCPX

	XL	VL	LO	NM	HI	VH	XH
Suma de RELY, DATA, CPLX, DOCU	5,6	7,8	9-11	12	13-15	16-18	19-21

- **RUSE** (Reusabilidad) es el mismo multiplicador RUSE del modelo Post-Arquitectura.
- **PDIF** (Dificultad de la plataforma) está dado por la combinación de los multiplicadores TIME, STOR y PVOL.

Tabla 2.58: Calificación PDIF

	LO	NM	HI	VH	XH
Suma de TIME, STOR, PVOL	8	9	10-12	13-15	16,17

- **PREX** (Experiencia del personal) combina los siguientes multiplicadores AEXP, PEXP y LTEX.

Tabla 2.59: Calificación PREX

	XL	VL	LO	NM	HI	VH	XH
Suma de AEXP, PEXP, LTEX	3,4	5,6	7,8	9	10,11	12,13	14

- **SCED** (Calendario) es el mismo multiplicador del modelo Post-Arquitectura.
- **FCIL** (Facilidades) combina los factores TOOL y SITE

Tabla 2.60: Calificación FCIL

	XL	VL	LO	NM	HI	VH	XH
Suma de TOOL y SITE	2	3	4,5	6	7,8	9,10	11

En la siguiente tabla se muestra los valores que toman los drivers de costo para el modelo de Diseño Inicial:

Tabla 2.61: Valores de Drivers de Costo para el Modelo de Diseño Inicial

Factores de Escala	XL	VL	LO	NM	HI	VH	XH
RCPX	0.73	0.81	0.98	1.00	1.30	1.74	2.38
RUSE			0.95	1.00	1.07	1.15	1.24
PDIF			0.87	1.00	1.29	1.81	2.61
PERS	2.12	1.62	1.26	1.00	0.83	0.63	0.50
PREX	1.59	1.33	1.12	1.00	0.87	0.71	0.62
FCIL	1.43	1.30	1.10	1.00	0.87	0.73	0.62
SCED		1.43	1.14	1.00	1.00	1.00	

Por ejemplo, si al analizar los drivers de costo para nuestro proyecto obtenemos los siguientes resultados:

Tabla 2.62: Datos del ejemplo

Factores de Escala	XL	VL	LO	NM	HI	VH	XH
RCPX					X		
RUSE				X			
PDIF				X			
PERS				X			
PREX					X		
FCIL					X		
SCED				X			

$$\pi EM_i = 1.30 \times 1 \times 1 \times 1 \times 1 \times 0.87 \times 0.87 \times 1 = 0.984$$

Cálculo del tiempo de desarrollo del proyecto

La ecuación siguiente muestra los tiempos base para las tres etapas de COCOMO II:

$$TDEV = \left[3.67 \times PM^{(0.28 + 0.2x(B-1.01))} \right] \times \frac{SCED\%}{100}$$

Donde:

PM es la estimación de mes-persona

B es la suma de los factores de escala del proyecto

SCED (calendario de Desarrollo expandido) es el porcentaje de compresión/expansión en el multiplicador de esfuerzo SCED.

Así por ejemplo, si tomamos un proyecto en el que hemos obtenido un resultado de 26 MM, B=0.75 y no existen restricciones de tiempo, al aplicar la ecuación del tiempo se obtendría:

$$TDEV = [3.67 \times 26^{(0.28+0.2 \times (0.75 - 1.01))}]$$

$$TDEV = 7.71$$

Comparación entre COCOMO 81 y COCOMO II

Tabla 2.63: Comparación entre COCOMO 81 y COCOMO II

COCOMO 81		COCOMO II
ESTRUCTURA DEL MODELO	Un modelo único que asume que se ha comenzado con unos requisitos asignados para el software	Tres modelos que asumen que se progresa a lo largo de un desarrollo de tipo espiral para consolidar los requisitos, la arquitectura y reducir el riesgo
FORMA MATEMÁTICA DE LA ECUACIÓN DEL ESFUERZO	$Esfuerzo = A(c_i) (SIZE)^{Exponente}$	$Esfuerzo = A(c_i) (SIZE)^{Exponente}$
EXPONENTE	Constante fija seleccionada como una función de modo: <ul style="list-style-type: none"> • Orgánico =1.05 • Semi-acoplado = 1.12 • Empotrado = 1.20 	Variable establecida en función de una medida de cinco factores de escala: <ul style="list-style-type: none"> • PREC Precedencia • FLEX Flexibilidad de desarrollo • RESL Resolución de Arquitectura • TEAM Cohesión del equipo

		<ul style="list-style-type: none"> • PMAT Madurez del proceso
MEDIDA	Líneas de código fuente (con extensiones para puntos de función)	Puntos objeto, Puntos de función o líneas de código fuente.
DRIVERS DE COSTE	<p>15 drivers, cada uno de los cuales debe ser estimado:</p> <ul style="list-style-type: none"> • RELY Fiabilidad • DATA Tamaño Base de datos • CPLX Complejidad • TIME Restricción tiempo de ejecución • STOR Restricción de almacenamiento principal • VIRT Volatibilidad máquina virtual • TURN Tiempo de respuesta • ACAP Capacidad del analista • PCAP Capacidad programador • AEXP Experiencia aplicaciones • VEXP Experiencia máquina virtual • LEXP Experiencia lenguaje • TOOL Uso de las herramientas software • MODP Uso de técnicas modernas de programación • SCED Planificación requerida 	<p>17 drivers, cada uno de los cuales debe ser estimado:</p> <ul style="list-style-type: none"> • RELY Fiabilidad • DATA Tamaño Base de datos • CPLX Complejidad • RUSE Reutilización requerida • DOCU Documentación • TIME Restricción tiempo de ejecución • STOR Restricción de almacenamiento principal • PVOL Volatibilidad plataforma • ACAP Capacidad del analista • PCAP Capacidad programador • AEXP Experiencia aplicaciones • PEXP Experiencia plataforma • LTEX Experiencia lenguaje y herramienta • PCON Continuidad del personal • TOOL Uso de las herramientas software • SITE Desarrollo multi-lugar • SCED Planificación requerida
OTRAS DIFERENCIAS DEL MODELO	<p>Modelo basado en:</p> <ul style="list-style-type: none"> • Fórmula de reutilización lineal • Asunción de requisitos razonables estables 	<p>Tiene muchas otras mejoras que incluyen:</p> <ul style="list-style-type: none"> • Fórmula de reutilización No lineal • Modelo de reutilización que considera esfuerzo necesario para entender y asimilar • Medidas de rotura que se usan

		para abordar la volatilidad de requisitos <ul style="list-style-type: none"> • Características de autocalibración
--	--	--

Después de haber entendido los parámetros y factores que permiten determinar el esfuerzo, el tiempo de desarrollo de un proyecto de software, lo llevaremos a un ejemplo práctico, con la ayuda de COCOMO USC.

Primero lo resolveremos aplicando los parámetros, factores y fórmulas anteriormente aprendidas. Una empresa ha estimado a desarrollar un sistema de aproximadamente 5000 líneas de código. Este software genera necesidades en cuantos a los factores de escala, se requiere estimar utilizando el método de COCOMO II Post-Arquitectura.

Cálculo de Esfuerzo

$$PM_{NOMINAL} = A * (SIZE)^B$$

$$A = 2.45$$

$$B = 1.01 + 0.01x \sum_{j=1}^5 SF_j$$

Los factores de escala considerados para este proyecto de software son:

Tabla 2.64: Factores de escala del ejercicio

Factores de Escala	VL	LO	NM	HI	VH	XH
PREC			X			
FLEX	X					
RESL		X				
TEAM	X					
PMAT	X					

Sumando sus valores tenemos:

$$= 3.72 + 5.07 + 5.65 + 5.48 + 7.80$$

$$= 27.72$$

$$B = 1.01 + 0.01 (27.72)$$

$$B = 1.29 = 1.3$$

$$PM_{NOMINAL} = 2.45 * (5)^{1.3}$$

$$PM_{NOMINAL} = 19.85$$

$$\begin{aligned} \text{Productividad} &= (KLOC / PM_{NOMINAL}) \\ &= 5000 / 19.85 \\ &= 251.89 \end{aligned}$$

Esfuerzo Nominal por Módulo $PM_{NOMINAL, MÓDULO}$ se obtiene como el cociente entre el tamaño del módulo (5000) y la Productividad del Proyecto

$$= 5000 / 251.89$$

$$= 19.85 = 19.9$$

Seguidamente determinar los niveles en que se encuentra cada factor de costo (Muy bajo, Bajo, Nominal, Alto, Muy alto) asignar los valores de los multiplicadores de esfuerzo correspondientes. Los 17 factores fueron considerados en un nivel nominal.

Tabla 2.65: Factores de Costo del ejercicio

RELY	DATA	CPLX	RUSE	DOCU	TIME	STOR	PVOL	ACAP	PCAP	PCON	AEXP	PEXP	LTEX	TOOL	SITE	SCED
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$$EM = 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1$$

$$EM = 1$$

$$\begin{aligned} \text{Esfuerzo estimado} &= PM_{NOMINAL, MÓDULO} \times EM \\ &= 19.9 \times 1 \\ &= 19.9 \end{aligned}$$

Determinar el Tiempo de Desarrollo

$$TDEV = \left[3.67 \times 19.9^{(0.28 + 0.2 \times (1.3 - 1.01))} \right] \times \frac{SCED\%}{100}$$

$$TDEV = 10.1$$

El Costo Mes-Persona es de 450 dólares y eso multiplicarlo por el Esfuerzo estimado

$$\text{Costo}_{\text{ESTIMADO}} = 450 \times 19.9 = \mathbf{8955}$$

El costo por instrucción en \$, se obtiene como el cociente entre el Costo de Desarrollo y el tamaño del módulo.

$$\text{Costo por instrucción} = 8955 / 5000 = 1.8 \text{ dólares}$$

SOFTWARE COCOMO USC

Para verificar que los cálculos anteriores son correctos nos ayudaremos con el Software COCOMO II.1999.0 USC. Se debe ingresar el nombre del Proyecto, módulo, el tamaño de líneas de código; el modelo a ser desarrollado es Post-Arquitectura y el número de líneas de código para nuestro caso de estudio es igual a 5000 KLOC.

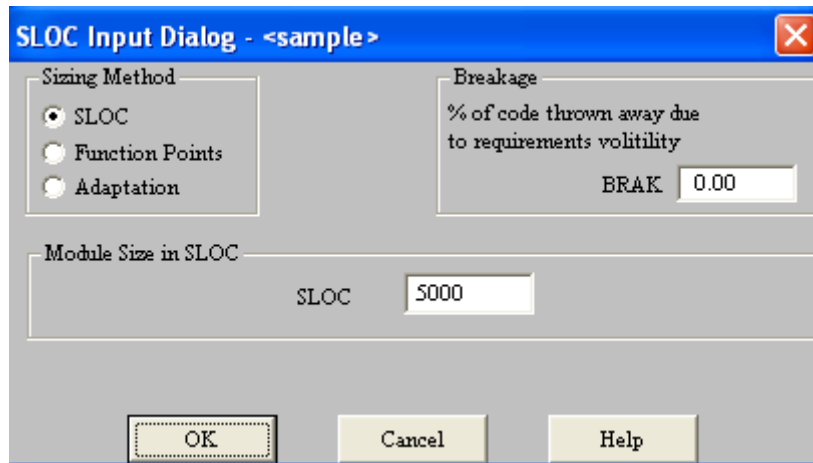


Figura 2.22: Ubicación de KLOC en el software

Ubicar la tarifa de trabajo en \$/mes, el cual corresponde a la cantidad de dinero que debe cobrar por mes el desarrollador que se dedica a cada módulo, se ha considerado 450 dólares.

Los factores de escala a ser aplicados a todo el proyecto PREC, FLEX, RESL, TEAM, PMAT serán:

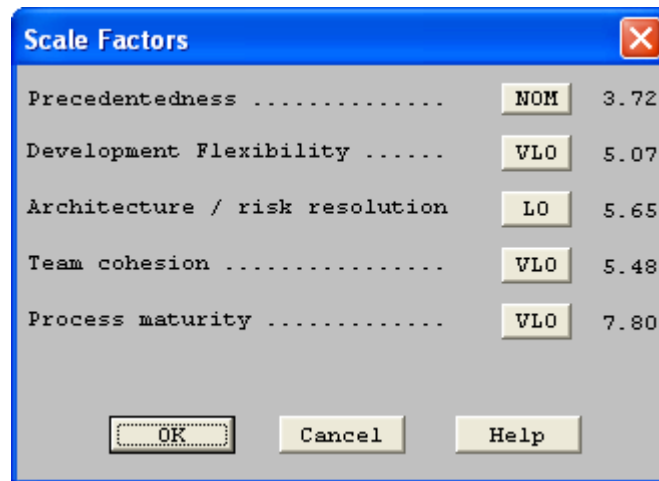


Figura 2.23: Factores de escala en el software

Obteniendo los siguientes resultados:

Module Size	LABOR Rate (\$/month)	ERF	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
S: 5000	450.00	1.00	19.9	19.9	251.7	8940.81	1.8	2.0	0.3

Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Optimistic	15.9	9.3	314.6	7152.65	1.4	1.7	
Most Likely	19.9	10.0	251.7	8940.81	1.8	2.0	0.3
Pessimistic	24.8	10.8	201.3	11176.02	2.2	2.3	

Figura 2.24: Resultados

Tamaño total del proyecto en líneas de código (Total Lines of Code)=5000

Las estimaciones Optimista (Optimistic), más probable (Most Likely) y pesimista (Pessimistic) indican el esfuerzo (Effort), la duración (Sched), la productividad (PROD), el coste (COST), el coste por instrucción (INST), el personal necesario (Staff) y el riesgo (RISK) de nuestro caso de estudio en cada uno de éstos parámetros.

Personal necesario (Staff) es igual a 2 desarrolladores para completar un módulo en el tiempo de desarrollo probable de estimación.

Coste por instrucción (INST COST) es igual a 1.8 dólar por línea de código.

Coste (COST) = 8940 dólares para desarrollar el módulo.

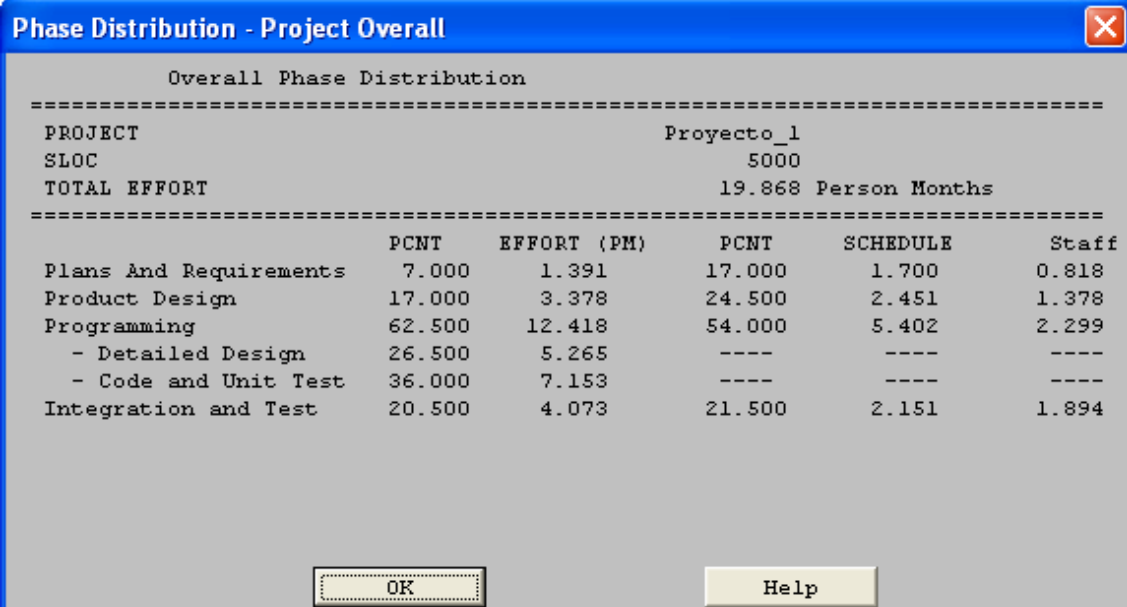
Productividad (PROD) = 251.7

Riesgo (RISK) = 0.3 siendo esto el nivel de riesgo total del módulo.

Algunos datos adicionales que le presenta el programa se muestran a continuación:

Proyecto Total

El sistema le permite visualizar tanto el esfuerzo como el tiempo de desarrollo completo del proyecto distribuido por fase. Según como se muestra en el gráfico 5.4 y 5.5. En la primera columna se define el nombre de la fase. Segunda columna porcentaje respectivo. Tercera columna esfuerzo estimado. Cuarta columna porcentaje del esfuerzo, quinta columna el porcentaje del tiempo estimado que se dedicaría a cada una de ellas. La última columna exhibe el número de personas necesarias para la terminación de la fase.



The screenshot shows a window titled "Phase Distribution - Project Overall" with a close button in the top right corner. The window displays the following text:

```
Overall Phase Distribution
=====
PROJECT                               Proyecto_1
SLOC                                   5000
TOTAL EFFORT                           19.868 Person Months
=====
```

	PCNT	EFFORT (PM)	PCNT	SCHEDULE	Staff
Plans And Requirements	7.000	1.391	17.000	1.700	0.818
Product Design	17.000	3.378	24.500	2.451	1.378
Programming	62.500	12.418	54.000	5.402	2.299
- Detailed Design	26.500	5.265	----	----	----
- Code and Unit Test	36.000	7.153	----	----	----
Integration and Test	20.500	4.073	21.500	2.151	1.894

At the bottom of the window, there are two buttons: "OK" and "Help".

Figura 2.25: Distribución del esfuerzo y tiempo de desarrollo del sistema total por fase

Diseño del Producto

En esta etapa muestra el esfuerzo requerido para cada fase con respecto al diseño del producto.

Phase Distribution - Project Product Design				
=====				
Life Cycle Phase	Product Design			
Life Cycle Effort	3.378 Person Months			
Life Cycle Schedule	2.451 Months			
=====				
	PCNT	EFFORT (PM)	SCHEDULE	Staff
Requirements Analysis	12.500	0.422	2.451	0.172
Product Design	41.000	1.385	2.451	0.565
Programming	12.250	0.414	2.451	0.169
Test Planning	4.750	0.160	2.451	0.065
Verification and Validation	6.250	0.211	2.451	0.086
Project Office	12.500	0.422	2.451	0.172
CM/QA	2.750	0.093	2.451	0.038
Manuals	8.000	0.270	2.451	0.110

OK Help

Figura 2.26: Diseño del Producto Software

Programación del Proyecto

El software COCOMO II realiza una programación del esfuerzo y el tiempo considerando las fases de requerimientos de análisis, diseño del producto, programación, pruebas, verificación/validación y manuales.

Phase Distribution - Project Programming				
=====				
Life Cycle Phase	Programming			
Life Cycle Effort	12.418 Person Months			
Life Cycle Schedule	5.402 Months			
=====				
	PCNT	EFFORT (PM)	SCHEDULE	Staff
Requirements Analysis	4.000	0.497	5.402	0.092
Product Design	8.000	0.993	5.402	0.184
Programming	56.500	7.016	5.402	1.299
Test Planning	4.250	0.528	5.402	0.098
Verification and Validation	7.250	0.900	5.402	0.167
Project Office	7.250	0.900	5.402	0.167
CM/QA	6.750	0.838	5.402	0.155
Manuals	6.000	0.745	5.402	0.138

OK Help

Figura 2.27 Programación del Proyecto

Integración y prueba del proyecto

En esta ventana el usuario visualizará los componentes de esta fase en particular. El gráfico 5.8 exhibe la información siguiente: Nombre de módulo, el módulo total SLOC, el esfuerzo y el tiempo estimado total del módulo. Además muestra el análisis de requisitos de las actividades, diseño del producto, etc. Las estimaciones de cada actividad son acompañadas por el porcentaje del esfuerzo, el esfuerzo, el tiempo y el personal necesario para la actividad.

Phase Distribution - Project Integration & Test				
=====				
Life Cycle Phase	Integration and Test			
Life Cycle Effort	4.073 Person Months			
Life Cycle Schedule	2.151 Months			
=====				
	PCNT	EFFORT (PM)	SCHEDULE	Staff
Requirements Analysis	2.500	0.102	2.151	0.047
Product Design	5.000	0.204	2.151	0.095
Programming	34.000	1.385	2.151	0.644
Test Planning	2.500	0.102	2.151	0.047
Verification and Validation	31.500	1.283	2.151	0.597
Project Office	8.250	0.336	2.151	0.156
CM/QA	8.250	0.336	2.151	0.156
Manuals	8.000	0.326	2.151	0.152

Figura 2.28 Pruebas del Proyecto Software

2.3.6 LECCIÓN No. 6: Planificación de Proyectos

Identificar actividades o tareas

Lo primero que debemos hacer dentro de la planificación de proyectos es identificar un conjunto de tareas, hitos y entregas que deben realizarse para completar el proyecto.

El conjunto de tareas a elegir debe proporcionar suficiente disciplina para alcanzar una alta calidad para el software.

Los hitos son puntos finales de una actividad o tarea del proceso de software. En cada uno, debe existir una salida formal, como un informe. Los hitos deben representar el fin de una etapa lógica en el proyecto. Para establecer los hitos, se debe dividir el proceso de

software en actividades básicas con sus salidas asociadas. Así como lo muestra el gráfico 2.47

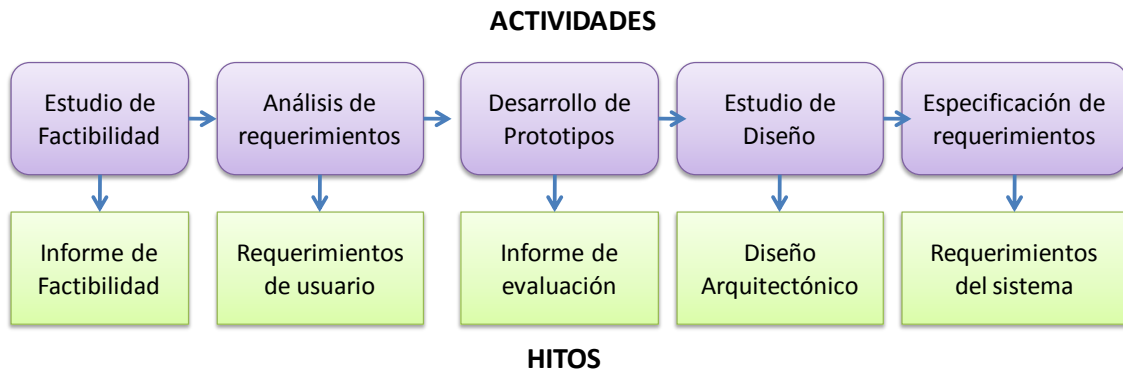


Figura 2.29: Hitos en el proceso de requerimientos

Las tareas que hemos identificado que se van a realizar durante el proceso de desarrollo para obtener la aplicación son:

1. Análisis y especificación de requisitos
2. Modelo Conceptual
3. Diseño Arquitectónico
4. Diseño Detallado
5. Codificación y pruebas

Estimación de los tiempos y costes de las actividades

Una vez que se ha definido la secuencia entre las actividades, es necesario realizar una estimación del tiempo que debe transcurrir entre el comienzo y el final de la actividad.

Estas estimaciones no se basan sólo en el esfuerzo en horas/hombre de una actividad, sino especialmente en el tiempo requerido para finalizar una actividad. Tales estimaciones suelen estar basadas en la experiencia del planificador en proyectos similares y deben incluir los retrasos normales. Es una buena estrategia para el jefe de proyecto contrastar y negociar sus cálculos de tiempos y esfuerzos con los responsables de cada actividad, que suelen contar con buena precisión a la hora de estimar los costes reales en tiempo y esfuerzo.

El costo del proyecto ha sido calculado a través de las técnicas de estimación aprendidas en la lección anterior, para nuestro proyecto de software el Costo del Proyecto es igual a 8955 dólares.

Distribución del tiempo

Normalmente las actividades del proyecto de software debe durar por lo menos una semana. Los administradores deben estimar los recursos necesarios para completar cada tarea. El recurso principal es el esfuerzo humano que se requiere, y el cual fue calculado en la lección anterior. Como regla se ha determinado agregar un 30% a la estimación inicial para problemas que podrían presentarse y otro 20% para cubrir algunas cosas no previstas.

Por lo general, el calendario se representa como un conjunto de gráficos que muestran la división del trabajo, las dependencias de las actividades y la asignación del personal.

Microsoft Project permite automatizar la producción de diagramas, en las cuales todas las tareas del proyecto se listan en la columna izquierda. Las barras horizontales indican la duración de cada tarea. Cuando aparecen múltiples barras al mismo tiempo en la planificación implica concurrencia de tareas. Los rombos indican los hitos.

En nuestra estimación, el proyecto tendrá una duración de 10 meses (160 días laborables) pero tomando en cuenta la recomendación de holgura se estableció planificar el proyecto a 15 meses (240 días).

Asignación de recursos

La asignación de recursos pretende determinar qué recursos serán necesarios, cuándo, cómo y dónde se obtendrán los que no están disponibles y en qué forma serán generados o adquiridos.

Se debe tener en cuenta cinco tipos de recursos:

- Los insumos (materiales, piezas, energía y servicios);
- Las instalaciones y equipo (inversiones de capital);
- El personal;

- La información;
- El dinero.

La herramienta principal para la asignación de recursos es el presupuesto; y éste se compone de la asignación de responsabilidades para generar y utilizar el dinero, y del calendario para hacerlo.

A continuación se muestra la asignación del recurso personal, según nuestra estimación se requiere de dos desarrolladores los cuales serán Jane y Anny, Jefe de proyecto será Richard, Analista es Freddy, Eduardo arquitecto.

Tabla 2.66: Recurso Personal

Recurso	Rol	Actividad
Richard	Jefe de Proyecto y revisor del diseño	Estudio previo, pruebas de aceptación
Freddy	Analista	Construcción del modelo de objetos
Jane	Desarrollador	Identificación de requisitos, implementación de escenarios
Anny	Desarrollador	Implementación de escenarios
Eduardo	Arquitecto	Análisis de Arquitectura y Diseño de Arquitectura

Elaboración del plan

El plan de proyecto de software se produce en el momento que se culmina las tareas de planificación, proporcionando información básica de costos y la planificación que será empleada a lo largo del proceso de la ingeniería de software. Su propósito es ayudar a establecer la viabilidad del esfuerzo de desarrollo de software.

El contenido del plan podría ser el siguiente:

Tabla 2.67: Plan de Proyecto de Software

PLAN DE PROYECTO DE SOFTWARE
I. Introducción

- A. Propósito del plan
- B. **Ámbito del proyecto y objetivos**
 - a. Declaración del ámbito
 - b. Funciones principales
 - c. Aspectos de rendimiento
 - d. Restricciones técnicas y de gestión
- II. Estimaciones del proyecto
 - A. Datos históricos usados para las estimaciones
 - B. Técnicas de estimación
 - C. Estimaciones de esfuerzo, costo y duración
- III. **Estrategia de riesgo**
 - A. Tabla de riesgo
 - B. Estudio de los riesgos a tratar
 - C. Plan RSGR
- IV. Planificación temporal
 - A. Estructura de descomposición del trabajo
 - B. Tareas
 - C. Gráfico de tiempo
 - D. Tabla de recursos
- V. Recursos del proyecto
 - A. Personal
 - B. Hardware y software
 - C. Tabla de recursos
- VI. Organización del personal
 - A. Estructura de equipo
 - B. Informes de gestión
- VII. Mecanismos de seguimiento y control
 - A. Garantía de calidad y control
 - B. Gestión de control y cambios
- VIII. Apéndices

← Éste contenido se aprenderá en la lección 7

Definición de hitos de control

Luego de saber las tareas de cada fase, se debe determinar hitos de control:

Tabla 2.68: Hitos del proyecto

No. Hito	Descripción
Hito 1	Entrega de la Especificación IEEE
Hito 2	Modelo de Objetos en UML
Hito 3	Diseño detallado del Caso de Estudio
Hito 4	Implementación de la carga
Hito 5	Implementación de escenarios
Hito 6	Entrega de la aplicación

Calendarización

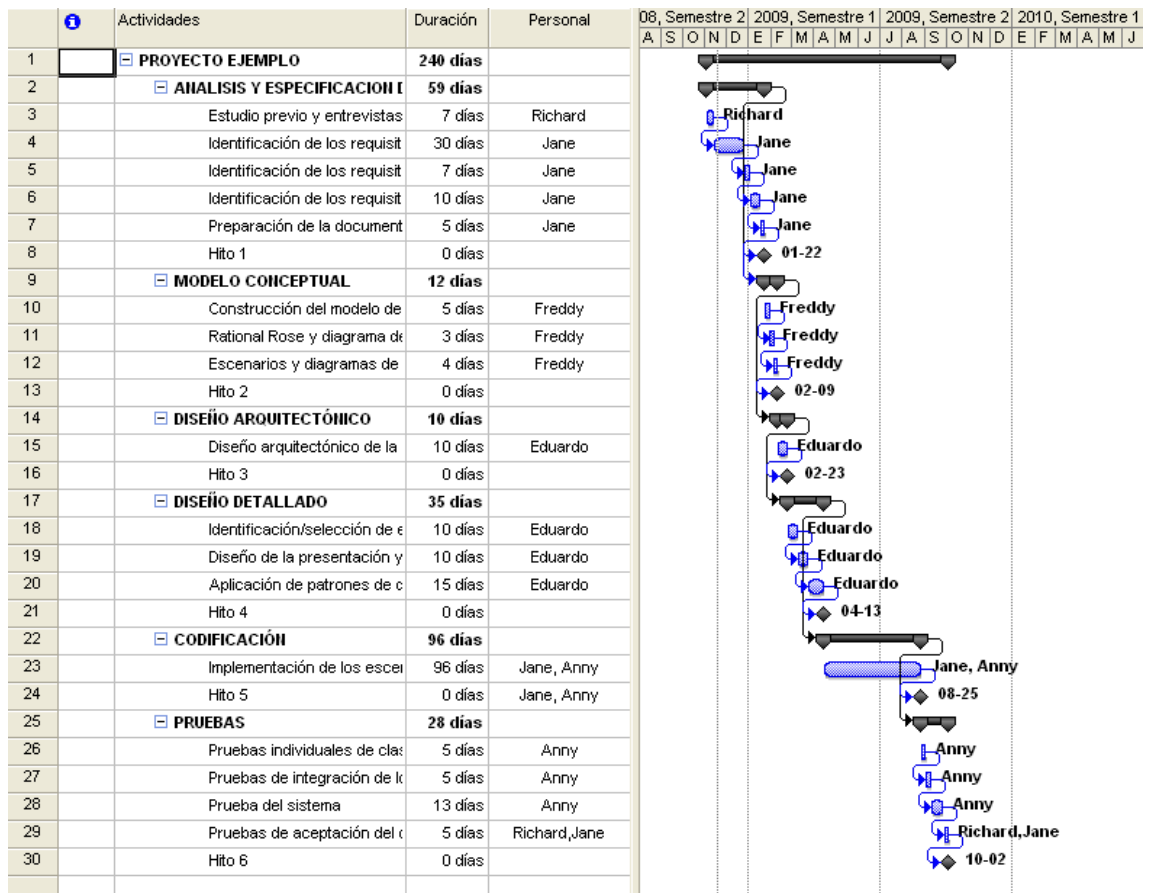


Figura 2.30: Tabla de Tareas

Gestión de Proyectos

Control de un proyecto

Controlar es el conjunto de actividades de gestión utilizadas para asegurar que el proyecto va de acuerdo a lo planificado. El desempeño y los resultados se miden contra los planes, se notan las desviaciones, y se toman acciones correctivas.

Actividades de control:

- Desarrollar estándares de desempeño
- Establecer sistemas de monitoreo y reportes
- Medir y analizar resultados
- Iniciar acciones correctivas
- Recompensar y disciplinar
- Documentar los métodos de control.

Como ejemplo tomaremos las siguientes áreas de conocimiento y se determinará su control:

Tabla 2.69: Control de áreas de conocimiento

Área de Conocimiento	Control	Indicador
Integración	Control integral de cambios	$\frac{\# \text{ cambios solicitados}}{\# \text{ cambios planificados}}$
Tiempo	Control del cronograma	$\frac{\text{Actividades ejecutadas}}{\text{Actividades planificadas}}$
Costo	Control del costo	$\frac{\text{Presupuesto ejecutado}}{\text{Presupuesto programado}}$
Calidad	Control de calidad	$\frac{\# \text{ pruebas realizadas}}{\# \text{ pruebas planificadas}}$
Comunicación	Información de desempeño	$\frac{\# \text{ PF individual}}{\# \text{ PF grupal}}$
Riesgo	Monitoreo y control de riesgo	$\frac{\# \text{ personas capacitadas}}{\# \text{ personas laborando}}$

Supervisión y revisión del proyecto

Es importante realizar un seguimiento del proyecto, de manera que sabremos en todo momento, si vamos cumpliendo el plan, o si nos estamos adelantando o retrasando. Este seguimiento se puede realizar gracias a una buena planificación de puntos de control. Esto permitirá averiguar en dónde hay problemas.

El seguimiento también permite detectar los problemas lo más pronto posible, para poder resolverlos antes de que se compliquen más.

Algunas características que se deben tomar en consideración al momento de revisar la planificación es:

- Los puntos de seguimiento deben ser claros
- No cambiar la programación hasta hacer un nuevo plan
- No cambiar el plan al informar de su situación real
- Guardar copias de la planificación original y de sus actualizaciones
- Si se realiza una estimación de nuevas fechas, no borrar la estimación anterior.

Acciones correctivas

Se determinan acciones correctivas en el caso de que los valores actuales del proyecto en cuanto a tamaño, esfuerzo, costo, calendarios de las actividades y expectativas de riesgo excedan considerablemente de los valores previstos.

Algunos ejemplos de situaciones que podrían producir desviaciones con respecto a lo planificado son:

Tabla 2.70: Desviaciones

DESVIACIONES DE LA PLANIFICACIÓN	<ul style="list-style-type: none"> • Nadie tiene idea de cuándo va a finalizar el proyecto y la mayoría de la gente renuncia a intentar imaginarlo. • El producto software está lleno de defectos • Los miembros del equipo trabajan un número excesivo de horas. • El proyecto está a punto de ser cancelado porque las entregas no se producen a tiempo y tienen
---	--

	numerosos defectos.
--	---------------------

Los problemas que produzcan estas desviaciones se deben documentar en los informes de problemas. Las acciones correctivas efectivas que se pueden adoptar para corregir adecuadamente el desarrollo de un proyecto son:

Tabla 2.71: Acciones correctivas

ACCIONES CORRECTIVAS	<ul style="list-style-type: none"> • Reducir el tamaño del software de tal manera que se pueda construir dentro del tiempo y el esfuerzo especificado. • Incrementar la productividad del proceso centrándose en las mejores a corto plazo. • Enfrentarse al hecho de que el software no va a estar listo a tiempo, retrasando la planificación y continuando con el control del problema. • Omitir prestaciones, aumentando la productividad.
-----------------------------	--

2.3.7 LECCIÓN No. 7: Administración de Riesgos

El manejo de riesgos involucra la identificación del riesgo y la escritura de planes para minimizar el efecto de estos en el proyecto.

Un riesgo se relaciona con la probabilidad de que ocurra alguna circunstancia adversa al proyecto.

Hay tres clases principales de riesgos:

- Los **riesgos del proyecto** afectan a la planificación o a los recursos.
- Los **riesgos del producto** afectan a la calidad o al desempeño del software por desarrollarse.
- Los **riesgos del negocio** son aquellos que afectan a la organización que desarrolla el software.

Los tipos de riesgo que pueden afectar un proyecto dependen de éste y el entorno organizacional en el que se está desarrollando el mismo. Sin embargo, muchos riesgos son universales; la tabla 2.66¹ describe algunos de ellos.

Tabla 2.72: Riesgos posibles en el software

Riesgo	Tipo de Riesgo	Descripción
Rotación del personal	Proyecto	Personal con experiencia abandona el proyecto antes de que finalice
Cambio de administración	Proyecto	Habrà un cambio de administración organizacional con diferentes prioridades
No disponibilidad de hardware	Proyecto	El hardware esencial para el proyecto no será entregado a tiempo
Cambio de requerimientos	Proyecto y Producto	Habrà más cambios en los requerimientos que lo anticipado
Retrasos en la especificación	Proyecto y Producto	Las especificaciones de las interfaces esenciales no estarán a tiempo
Subestimación del tamaño	Proyecto y Producto	El tamaño del sistema se ha subestimado
Cambio de tecnología	Negocio	La tecnología fundamental sobre la que se construirà el sistema se sustituye por nueva tecnología
Competencia del producto	Negocio	Un producto competitivo se pone en venta antes de que el sistema se complete.

Existen varios modelos de Administración de Riesgos pero el más utilizado consta de 5 pasos los que comparten como actividades comunes las de documentación y comunicación

¹ Sommerville 6 Edición, Capítulo 4 Administración de proyectos



Figura 2.31: Proceso de Administración de Riesgo

Identificación de Riesgos

Comprende el descubrimiento de los posibles riesgos del proyecto. Esta identificación se lo puede llevar a cabo a través de un proceso de grupo utilizando un enfoque de lluvia de ideas o simplemente puede basarse en la experiencia del administrador. Para ayudar al proceso, se utiliza una lista de posibles tipos de riesgos. Estos tipos incluyen:

1. **Riesgos de tecnología**, éstos se derivan de las tecnologías de software o de hardware
2. **Riesgos de personas**, riesgos asociados con las personas en el equipo de desarrollo
3. **Riesgos organizacionales**, se derivan del entorno organizacional donde el software se está desarrollando.
4. **Riesgos de herramientas**, se derivan de herramientas CASE y de otro software de apoyo
5. **Riesgos de requerimientos**, Éstos se derivan de los cambios de requerimientos del cliente y el proceso de administrarlo.
6. **Riesgos de estimación**, se derivan de los estimados administrativos de las características del sistema

El resultado de este proceso es una larga lista de riesgos que podrían ocurrir y afectar el producto, proceso o negocio. En la Tabla 2.73 se da algunos ejemplos de riesgos posibles en cada una de las categorías:

Tabla 2.73: Riesgos y tipos de software

<p>Riesgos de Tecnología</p>	<ul style="list-style-type: none"> • Se trata de una tecnología nueva en la organización • Se requieren nuevos algoritmos o tecnología de I/O • Se debe interactuar con hardware nuevo • Se debe interactuar con software que no ha sido probado • Se debe interactuar con un B.D. cuya funcionalidad y rendimiento no ha sido probada • Es requerido un interface de usuario especializado
<p>Riesgos de personas</p>	<ul style="list-style-type: none"> • Es el mejor personal disponible • Tienen los miembros las técnicas apropiadas • Hay suficiente gente disponible • Está el personal comprometido en toda la duración del proyecto • Habrá parte del personal dedicado solamente en parte al proyecto • Tiene el personal las expectativas correctas del trabajo • Tiene el personal la necesaria formación • Puede la rotación del personal perjudicar el proceso de desarrollo
<p>Riesgos organizacionales</p>	<ul style="list-style-type: none"> • Fecha límite de entrega razonable • Número de clientes que usarán el producto • Numero de productos con los que deberá interactuar • Sofisticación del usuario final • Cantidad y calidad de la documentación a entregar al cliente • Límites legales y gubernamentales • Costes asociados al retraso en la entrega

	<ul style="list-style-type: none"> • Costes asociados a errores en el producto
Riesgos de Herramientas	<ul style="list-style-type: none"> • Es ineficiente el código generado por las herramientas CASE • Las herramientas CASE no se pueden integrar
Riesgos de Requerimientos	<ul style="list-style-type: none"> • Se proponen cambios en los requerimientos que requieren rehacer el diseño • Los clientes no comprenden el impacto de los cambios en los requerimientos
Riesgos de estimación	<ul style="list-style-type: none"> • El tiempo requerido para desarrollar el software está subestimado. • La tasa de reparación de defectos está subestimada. • El tamaño del software está subestimado.

Análisis de Riesgos

Durante este proceso, se considera por separado cada riesgo identificado y se decide acerca de la probabilidad y la seriedad del mismo. No se hace una valoración con números precisos sino en intervalos:

1. la probabilidad de que el riesgo se valore como muy bajo (<10%), bajo (10-25%); moderado (25-50%), alto (50-75%) o muy alto (>75%).
2. los efectos de riesgo pueden ser valorados como catastrófico, crítico. Marginal y despreciable.

El resultado de este análisis se debe colocar en una tabla, la cual debe estar ordenada acorde a la seriedad del riesgo. En la Tabla 2.74 se ilustra riesgos identificados como ejemplo.

Tabla 2.74: Análisis de Riesgo

Riesgo	Probabilidad	Efectos
Los problemas financieros de la organización tienden a reducir el presupuesto del proyecto	Baja	Catastrófico
Es imposible reclutar personal con la	Alta	Catastrófico

experiencia requerida en el proyecto		
El personal clave está enfermo y no disponible en momentos críticos	Moderada	Serio
Los componentes de software a reutilizarse contienen defectos que limitan su funcionalidad	Moderada	Serio
Se proponen cambios en los requerimientos que requieren rehacer el diseño	Moderada	Serio
La base de datos que se utiliza no puede procesar muchas transacciones por segundo	Moderada	Serio
El tiempo requerido para desarrollar el software está subestimado	Alta	Serio
Las herramientas CASE no se pueden integrar	Alta	Tolerable
Los clientes no comprenden el impacto de los cambios en los requerimientos	Moderada	Tolerable
La capacitación para el personal no está disponible	Moderada	Tolerable
El tamaño del software está subestimado	Alta	Tolerable

Una vez que los riesgos se hayan analizado y clasificado, se debe discernir cuáles son los más importantes que se debe considerar durante el proyecto. Este discernimiento depende de la combinación de la probabilidad del riesgo en cuestión y los efectos del mismo. En general, se debe tomar en cuenta todos los riesgos catastróficos; así como los riesgos serios que tienen más probabilidad de ocurrir.

Boehm (1988) recomienda identificar y supervisar los “10 riesgos más altos”, el número de riesgos a supervisar dependerá del proyecto. Pueden ser 5 o 15, éste número debe ser manejable un número muy grande de riesgos requiere mucha información, para nuestro caso de estudio en esta lección e identificados en la tabla 7.2, se consideró los 7 que tienen consecuencias serias o catastróficas.

Planeación de Riesgos

Este proceso considera cada uno de los riesgos claves identificados y las estrategias para administrarlo.

Estas estrategias caen en tres categorías:

1. **Estrategias de anulación;** seguir éstas significa reducir la probabilidad de que ese riesgo surja. Un ejemplo es la estrategia para abordar los componentes defectuosos.
2. **Estrategias de disminución;** seguir éstas significa reducir el impacto del riesgo. Un ejemplo es la estrategia para abordar las enfermedades del personal
3. **Planes de contingencia;** seguir éstas significa que, si sucede lo peor, se está preparado para ello y se cuenta con una estrategia para abordarlo. Un ejemplo es la estrategia para los problemas financieros de la organización.

Las estrategias para nuestro ejemplo se muestran en la Tabla 2.75.

Tabla 2.75: Estrategias de Administración de Riesgos.

Riesgo	Estrategia
Problemas financieros de la organización	Preparar un documento breve para el Administrador que muestre que el proyecto hace contribuciones importantes a las metas del negocio.
Problemas de reclutamiento	Informar al cliente las dificultades potenciales y las posibilidades de retraso.
Enfermedad del personal	Reorganizar el equipo
Componentes defectuosos	Reemplazar los componentes defectuosos con los comprados de fiabilidad conocida.
Cambios en los requerimientos	Observar con detenimiento la información para valorar el impacto de los requerimientos
Desempeño de la Base de Datos	Investigar la posibilidad de comprar una base de datos con alto desempeño
Tiempo de desarrollo subestimado	Investigar los componentes comprados y la utilización de un generador de programas

Estrategias frente al riesgo

Estrategias reactivas.- son aquellas que se dan cuando se deja que los riesgos produzcan sus efectos (en este momento ya no es un riesgo, es una realidad) y entonces se actúa en consecuencia.

En estas condiciones lo único que cabe es tomar medidas correctoras (apagar incendios), lo que produce muchos tiempos perdidos, retrasos en el proyecto, gabinetes de crisis, etc.

Las estrategias reactivas no son aconsejables porque ponen en grave peligro el proyecto.

Estrategias proactivas.- pasan por la evaluación previa y sistemática de todos los riesgos inherentes al proyecto, evaluando sus consecuencias.

Esto produce la creación de un Plan de Gestión de Riesgos, con sus planes de evitación, minimización de consecuencias, planes de contingencia, etc.

En estas condiciones el objetivo es la evasión del riesgo, con menor tiempo de reacción frente a los efectos negativos y una mejor gestión del proyecto en su conjunto: menor tiempo y menor coste.

Supervisión de Riesgos

Esta etapa comprende el valor de cada uno de los riesgos identificados para decidir si éste es más o menos probable y cuándo los efectos del mismo han cambiado. Es cierto que esto no se puede observar de forma directa, por lo que se tiene que buscar otros factores para dar indicios de la probabilidad del riesgo y sus efectos. Estos factores dependen de los tipos de riesgo, algunos ejemplos que ayudan en la valoración de éstos tipos de riesgo se muestra en la tabla 2.70.

Los factores de riesgo son los elementos que determinan el riesgo. Su medición y monitorización determina el grado de acercamiento al riesgo.

Por cada riesgo es necesario identificar los factores que lo determinan. Un riesgo puede ser, a su vez, factor de riesgo en otros riesgos.

La supervisión de riesgos debe ser un proceso continuo y, en cada revisión del progreso de la administración, cada uno de los riesgos clave debe ser considerado por separado y discutido por la audiencia.

Tabla 2.76: Factores de Riesgo

Tipo de riesgo	Indicadores Potenciales
Tecnología	Entrega retrasada del hardware o de la ayuda al software, problemas tecnológicos reportados.
Personas	Baja moral del personal, malas relaciones entre los miembros del equipo, disponibilidad de empleo.
Organizacional	Chismes organizacionales, falta de acciones por el administrador principal.
Herramientas	Rechazo de los miembros del equipo para utilizar herramientas, quejas acerca de las herramientas CASE, peticiones de estaciones de trabajo más potentes.
Requerimientos	Peticiones de muchos cambios en los requerimientos, quejas del cliente.
Estimación	Fracaso en el cumplimiento de los tiempos acordados y en la eliminación de defectos reportados.

El seguimiento de riesgos de nuestro proyecto software, debe estar plasmado en el calendario de desarrollo que se elaboró en la lección anterior, como el ciclo de vida elegido es cascada su revisión de riesgo se lo realizará en la etapa de análisis y diseño global, la figura 2.32 muestra tareas de supervisión de riesgo:

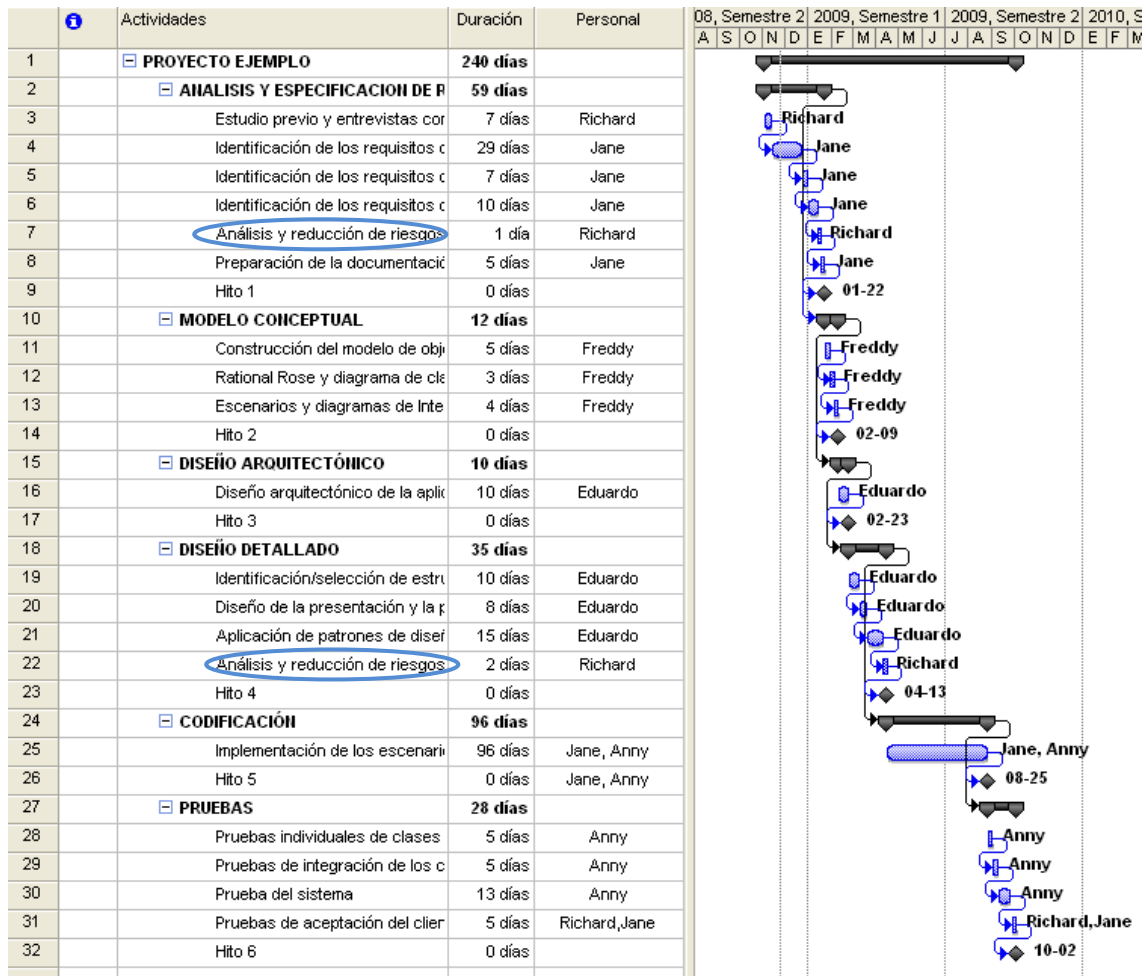


Figura 2.32: Calendario de proyecto de software con Gestión de Riesgos

El plan RSGR

Se puede incluir una estrategia de gestión de riesgo en el plan del proyecto de software o se podrían organizar los pasos de gestión del riesgo en un plan diferente de reducción, supervisión y gestión del riesgo (Plan RSGR). Todos los documentos del plan RSGR se llevan a cabo como parte del análisis de riesgo y son empleados por el jefe del proyecto como parte del Plan del Proyecto general. A continuación se expone un esquema del Plan RSGR.

Plan RSGR

I. Introducción

1. Alcance y propósito del documento
2. Visión general de los riesgos principales
3. Responsabilidades

- a. Gestión
 - b. Personal técnico
- II. Tabla de riesgo del proyecto.
- 1. Descripción de todos los riesgos por encima de la línea de corte
 - 2. Factores que influyen en la probabilidad e impacto
- III. Reducción, supervisión y gestión del riesgo
- 1. Reducción
 - a. Estrategia general
 - b. Pasos específicos
 - 2. Supervisión
 - a. Factores a supervisar
 - b. Enfoque de supervisión
 - 3. Gestión
 - a. Plan de contingencia
 - b. Consideraciones especiales
- IV. Planificación temporal de revisión del Plan RSGR

CAPITULO III

DESARROLLO DEL CURSO E-LEARNING DE LA MATERIA DE INGENIERÍA DE SOFTWARE I UTILIZANDO LA METODOLOGÍA OOHDM

OOHDM es una metodología orientada a objetos que plantea un proceso de desarrollo de cuatro fases donde se combinan notaciones gráficas UML (Lenguaje de Modelado Unificado) con otras propias de la metodología.

Para nuestro tesis de grado se ha propuesto la utilización de esta metodología por tanto debemos seguir las etapas o fases que contiene esta metodología. OOHDM propone el desarrollo de aplicaciones hipermedia a través de un proceso formado de las etapas que son: Diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación.

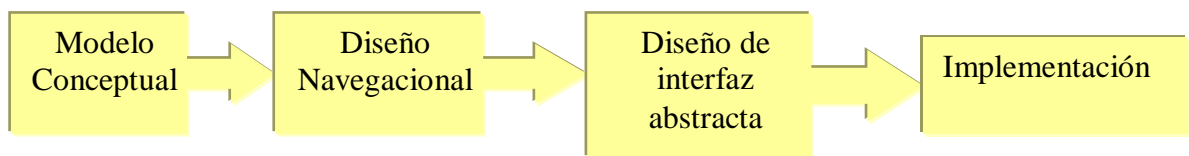


Figura 3.1. Etapas de la metodología OOHDM

3. ETAPAS

3.1.1. Diseño Conceptual

En esta actividad vamos a construir un esquema conceptual representado por los objetos del dominio, las relaciones y colaboraciones existentes entre ellos. En las aplicaciones hipermedia estándares, cuyos componentes no son modificados durante la ejecución, se podría usar un modelo de datos semánticas estructurales, de este modo en los casos de uso en que la información pueda cambiar dinámicamente se intente ejecutar cálculos complejos, se necesitará mejorar el comportamiento del modelo de objeto.

En el proyecto estará constituido por clases, relaciones y subsistemas, las clases son definidas como en los modelos orientados a objetos.

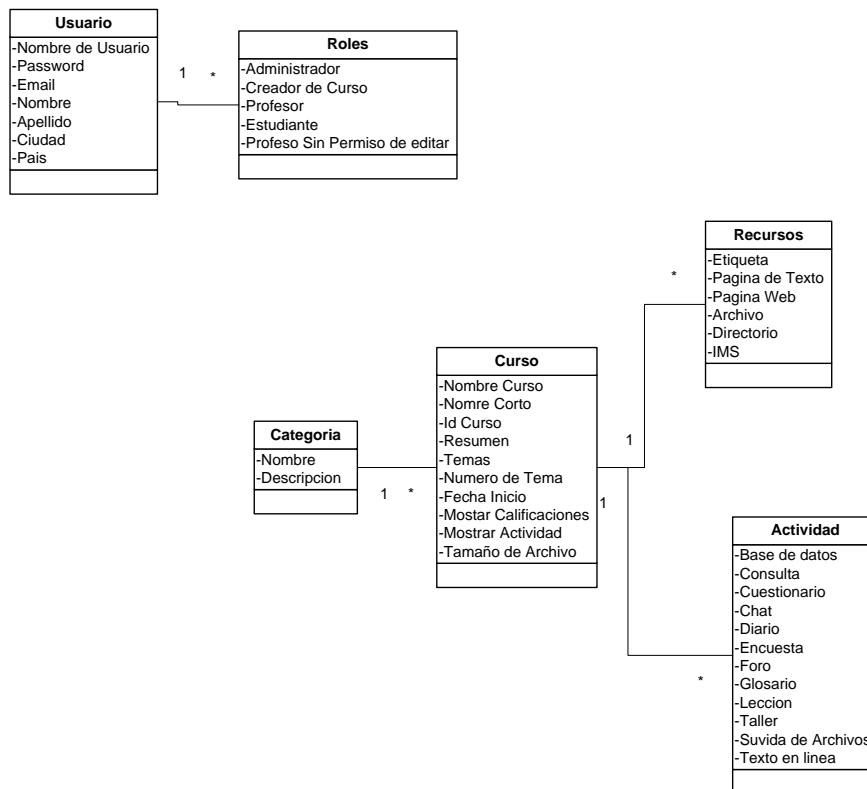


Figura 3.2 Mapa conceptual de un curso

3.1.2. Mapas de Navegación

En esta etapa de la metodología se pretende desarrollar una topología navegacional que permita a la aplicación (Moodle) ejecutar todas las tareas requeridas por el usuario. La idea principal es mostrar e integrar una serie de tareas para obtener el diseño navegacional del sistema

Los mapas navegacionales considerados para nuestro curso de e-learning se muestran en los siguientes gráficos, y han sido considerados desde la entrada a la página, el alumno encontrará el curso, las lecciones con el contenido de la materia:

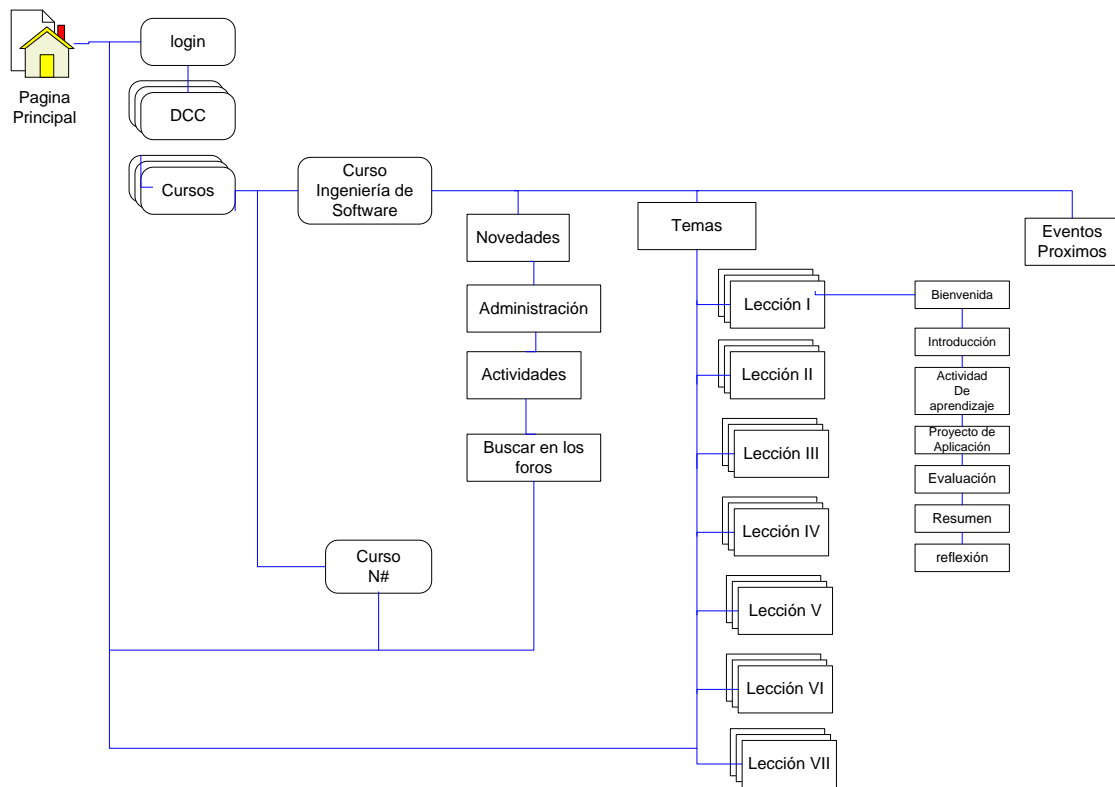


Figura 3.3. Mapa navegacional del Curso

3.1.3. Diseño de interfaces abstractas

En esta fase, se desarrolla un modelo orientado a objetos, para especificar la estructura y el comportamiento de la interfase del sistema hipermedia con el usuario, dicho modelo es abstracta y por tanto independiente de la implantación final del sistema

En nuestro proyecto no es necesario realizar esta fase, por cuanto la plataforma que utilizamos ya tiene definido su interfase, y al tratarse de una aplicación muy compleja, su interfase abstracta es muy difícil de demostrar en esta etapa, por consiguiente nuestra proyecto solo se dedica a la realización del curso y no de la plataforma.

3.1.4. Implementación

En esta ultima etapa o fase de la metodología OOHDH es la implementación del hiperdocumento o sistema hipermedia diseñado, es decir la concertación de los modelos navegacionales y de interfase en objetos particulares con sus correspondientes contenidos (presentaciones, texto, etc.) y sus posibilidades de navegación

En algunos textos de autores al modelo OOHDM se hace hincapié a una etapa más dentro de las cuatro mencionadas, esta etapa es la obtención de requerimientos que se realiza al iniciar el proyecto.

En esta etapa se realizan los estudios y se entregan los resultados claros y es en donde los errores que se cometes salen caros. Para realizar esta etapa se debe seguir algunas subetapas que son: Identificación de roles y tareas, Especificación de escenarios, Especificación de casos de uso, especificación de UIDs y validación de casos de uso y UIDs.

Pero como mencionamos anteriormente nuestro proyecto esta orientado a realizar el contenido del curso y no al desarrollo del sistema por tanto en esta etapa solo se realizará los casos de uso de algunas actividades.

3.1.5. Casos de uso

Casos de uso del sistema – Módulo Foros

Casos de uso de los módulos utilizados en el curso de ingeniería de software Las aportaciones de los usuarios en los foros, figura 3.4, permiten un intercambio asincrónico del grupo sobre un tema compartido. La participación en foros puede ser una parte integral de la experiencia de aprendizaje, ayuda a los alumnos a aclarar y desarrollar su comprensión del tema.

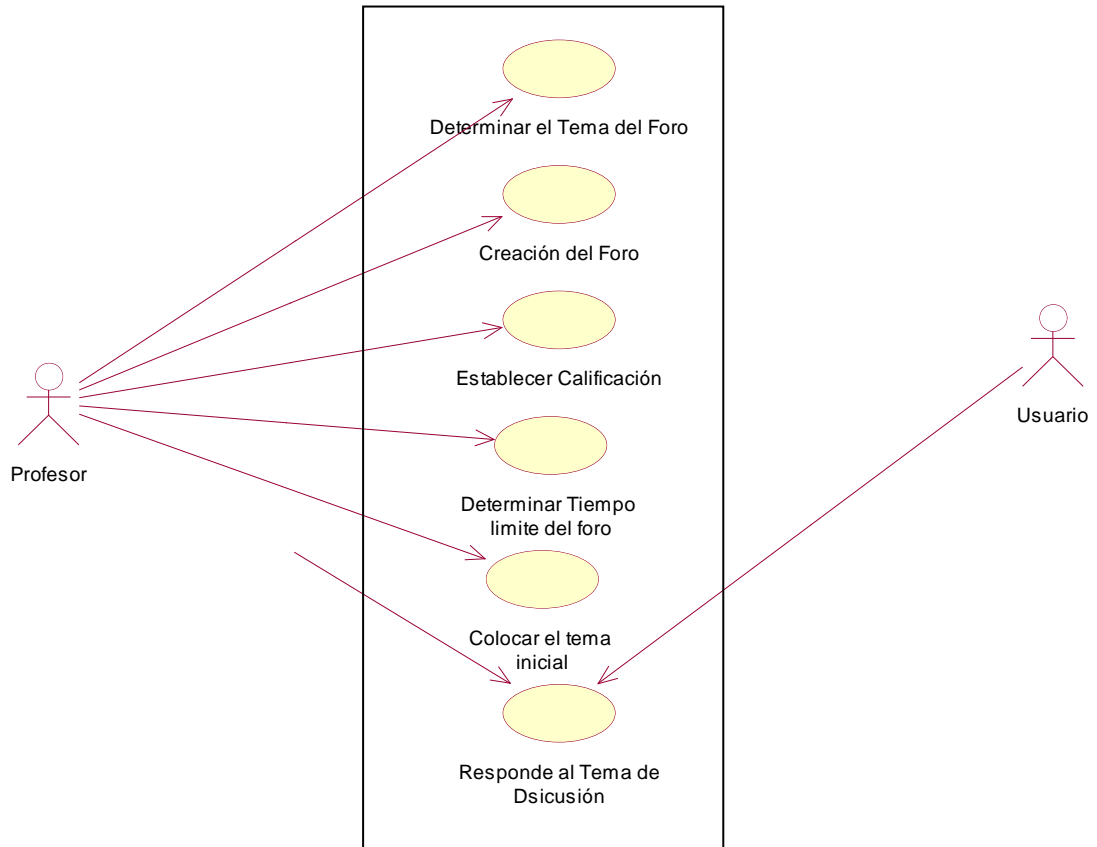


Figura 3.4 Casos de uso Foro

El módulo de Chat, figura 3,5, permite la comunicación en tiempo real de los usuarios.

Casos de uso del sistema – Módulo Chat

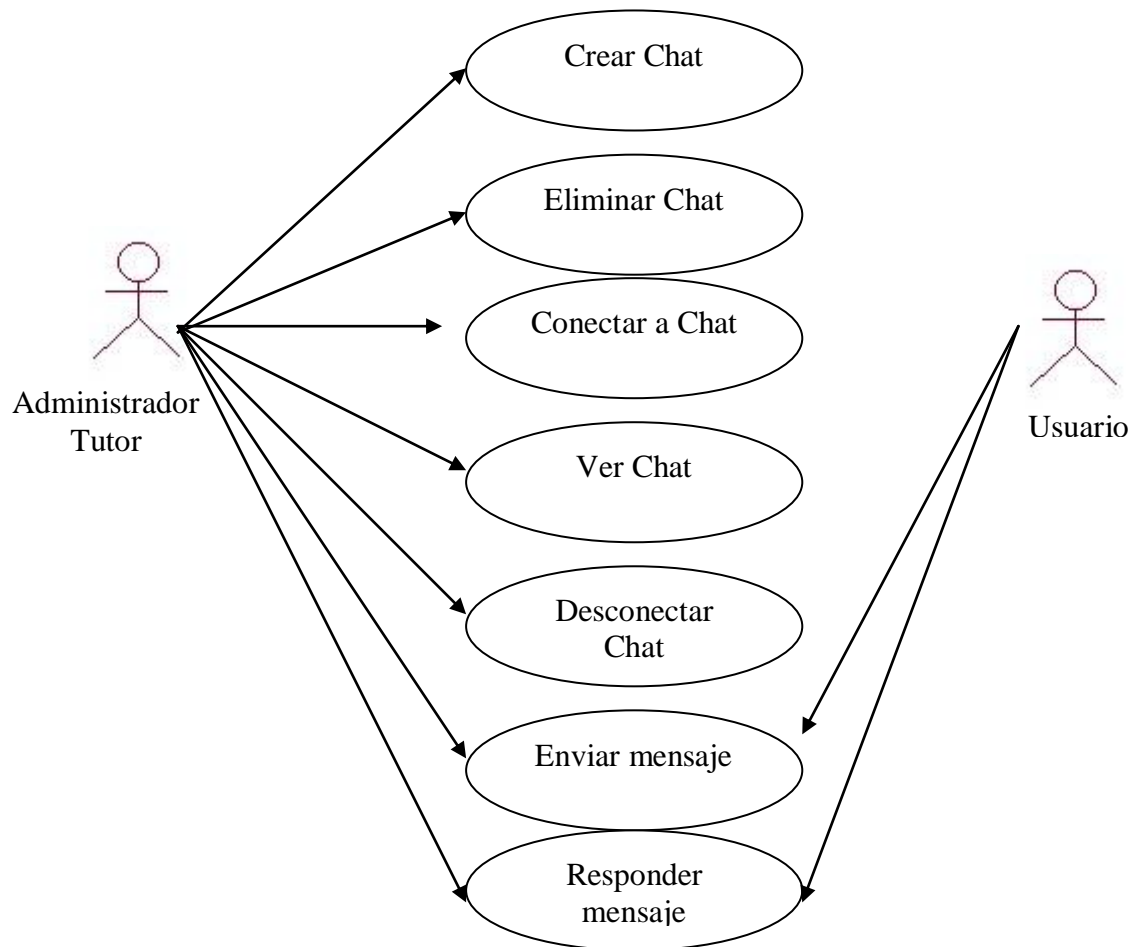


Figura 3.5 Casos de uso Chat

El módulo de recursos, figura 3.6 sirve para incluir contenidos en un curso; pueden ser: texto sin formato, archivos subidos, enlaces Web, Wiki o HTML o una referencia bibliográfica.

Casos de uso del sistema – Módulo Recursos

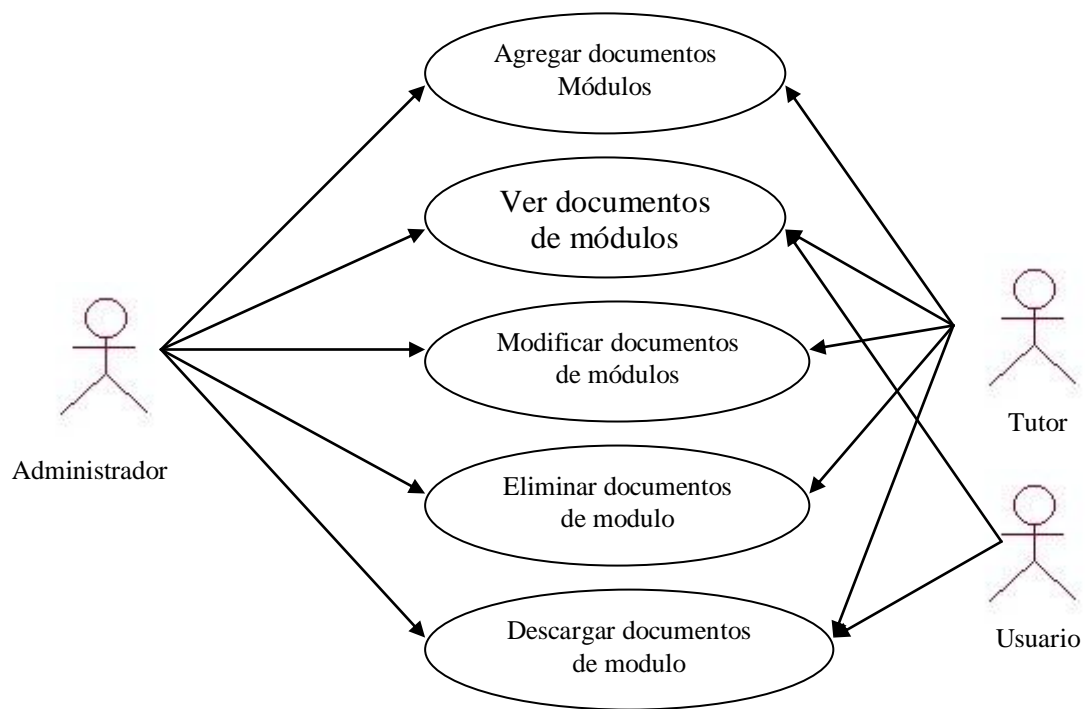


Figura 3.6 Casos de uso Recursos

Tabla 3.1 Especificación caso de uso – Ver Foro

Casos Real:	Ver Foro
Actor:	Administrador, tutor
Propósito:	Permitir al usuario ver el foro perteneciente a un modulo
Resumen:	1. El usuario selecciona la opción de foros

Tabla 3.2 Especificación caso de uso – Crear categoría o tipo de foro

Casos Real:	Crear categoría o tipo de foro
Actor:	Administrador, tutor
Propósito:	Permitir al tutor o administrador crear un tipo o categoría de foro de una lección
Resumen:	<ol style="list-style-type: none"> 1. El tutor pasa al modo de edición 2. El sistema muestra la interfase de modo edición 3. El usuario ingresa a la sección de agregar una actividad 4. El usuario selecciona la opción crear categoría o tipo de foro. 5. El usuario ingresa todos los campos necesario dentro del formulario de configuración. 6. El usuario selecciona la opción guardar cambios.

Tabla 3.3 Especificación caso de uso – Modificar categoría de foro

Casos Real:	Modificar categoría o tipo de foro
Actor:	Administrador, tutor
Propósito:	Permitir al tutor o administrador modificar un tipo o categoría de foro de una lección
Resumen:	<ol style="list-style-type: none"> 1. El tutor pasa al modo de edición 2. El sistema muestra la interfase de modo edición 3. El usuario ingresa a la sección de modificar una actividad 4. El usuario selecciona la opción cambiar categoría o tipo de foro. 5. El usuario ingresa todos los campos necesario dentro del formulario de configuración cambiar o modificar 6. El usuario selecciona la opción guardar cambios.

Tabla 3.4 Especificación caso de uso – Eliminar categoría de foro

Casos Real:	Eliminar foro
Actor:	Administrador, tutor
Propósito:	Permitir al tutor o administrador eliminar un foro o categoría
Resumen:	<ol style="list-style-type: none"> 1. El tutor pasa al modo de edición 2. El sistema muestra la interfase de modo edición 3. El Tutor/administrador ingresa a la gestión de actividades 4. El Tutor/administrador elige eliminar el foro

	5. El Tutor/administrador acepta o cancela la eliminación del foro
--	--

Tabla 3.5 Especificación caso de uso – Agregar mensaje o tema en foro

Caso Real:	Agregar tema o mensaje en el foro
Actor:	Administrador, Tutor , usuario
Propósito:	Permitir al usuario agregar un mensaje o tema de discusión en el foro
Resumen:	<ol style="list-style-type: none"> 1. El usuario ingresa al foro 2. El usuario pone un tema en foro 3. El usuario introduce un mensaje 4. El usuario coloca un archivo en foro 5. El usuario envía el tema al foro

Tabla 3.6 Especificación de caso de uso – Modificar mensaje o tema en foro

Caso Real:	Modificar mensaje en foro
Actor:	Administrador, Tutor, usuario
Propósito:	Permitir al usuario modificar un mensaje o tema enviado al foro
Resumen:	<ol style="list-style-type: none"> 6. El usuario ingresa al foro 7. El usuario lee el contenido del mensaje que va a modificar 8. El usuario guarda el mensaje modificado o archivo 9. El sistema muestra que el mensaje se modifico

Tabla 3.7 Especificación caso de uso – Eliminar un mensaje o tema en el foro

Caso Real:	Eliminar mensaje en foro
Actor:	Administrador, Tutor, usuario
Propósito:	Permitir al usuario eliminar un mensaje o tema enviado al foro
Resumen:	<ol style="list-style-type: none"> 1. El usuario ingresa al foro 2. El usuario lee el contenido del mensaje que va ha eliminar 3. El usuario elimina el mensaje y guarda los cambios 4. El sistema muestra que el mensaje se ha eliminado

Tabla 3.8 Especificación caso de uso – Crear Chat

Caso Real:	Crear Chat
Actor:	Administrador, Tutor
Propósito:	Permitir al Administrador/tutor crear un Chat en una lección
Resumen:	<ol style="list-style-type: none"> 1. El tutor pasa al modo de edición 2. El sistema muestra la interfase de modo edición 3. El Administrador/Tutor elige la opción agregar actividad 4. El Administrador/Tutor introduce la información y configuración del Chat. 5. El Administrador/Tutor guarda los cambios. 6. El Administrador/Tutor puede entra a la sal de Chat.

Tabla 3.9 Especificaciones de caso de uso – eliminar Chat

Caso Real:	Eliminar Chat
Actor:	Administrador, Tutor
Propósito:	Permitir al Administrador/tutor eliminar un Chat en una lección
Resumen:	<ol style="list-style-type: none"> 1. El tutor pasa al modo de edición 2. El sistema muestra la interfase de modo edición 3. El Administrador/Tutor elige la opción gestión de actividad 4. El Administrador/Tutor selecciona eliminar Chat 5. El Administrador/Tutor mensaje de confirmación de eliminación de Chat 6. El Administrador/Tutor puede entra a la sal de Chat.

Tabla 3.10 Especificaciones de caso de uso – Ver Chat

Caso Real:	Ver Chat
Actor:	Administrador, Tutor, usuario
Propósito:	Permitir al usuario ver la información referente al Chat de una lección
Resumen:	<ol style="list-style-type: none"> 1. El usuario ingresa al Chat 2. El sistema muestra la información referente al Chat de la lección selección.

Tabla 3.11 Especificaciones de caso de uso – Insertar/ enviar mensaje en el Chat

Caso Real:	Insertar Chat
Actor:	Administrador, Tutor, usuario
Propósito:	Permitir al usuario insertar un mensaje dentro del Chat
Resumen:	<ol style="list-style-type: none"> 1. El usuario ingresa al Chat 2. El usuario escribe el mensaje y presiona enter para enviar el mensaje 3. El mensaje es enviado y observado en la ventana de mensajes

Tabla 3.12 Especificación de caso de uso – Responder mensaje de Chat

Caso Real:	Responder Chat
Actor:	Administrador, Tutor, usuario
Propósito:	Permitir al usuario responder a mensaje enviado en cha de una lección.
Resumen:	<ol style="list-style-type: none"> 1. El usuario ingresa al Chat 2. El usuario lee el mensaje recibido del Chat 3. El usuario escribe y responde al mensaje.

Tabla 3.14 Especificaciones de caso de uso – Conectar Chat

Caso Real:	Conectar Chat
Actor:	Administrador, Tutor, usuario
Propósito:	Permitir al usuario ingresar a la selección de Chat de una lección
Resumen:	<ol style="list-style-type: none"> 1. El usuario ingresa al Chat 2. El sistema realiza la conexión al respectivo Chat de la lección 3. El sistema muestra la ventana relacionada al Chat.

Tabla 3.15 Especificaciones de caso de uso – desconectar Chat

Caso Real:	desconectar Chat
Actor:	Administrador, Tutor, usuario
Propósito:	Permitir al usuario salir de la sección de Chat de la lección.
Resumen:	<ol style="list-style-type: none"> 1. El usuario selecciona la opción salir del Chat de una lección. 2. El sistema realiza la desconexión.

Tabla 3.16 Especificación de casos de uso – Agregar recurso a lección o módulo

Caso Real:	Agregar recursos a la lección o módulo
Actor:	Tutor
Propósito:	Permitir al tutor agregar un recurso en una lección para el uso de los usuarios. En este propósito los administradores pueden ingresar recursos pero debido a que el tutor es el encargado o experto de la materia solo el es la persona idónea para crear este recurso.
Resumen:	<ol style="list-style-type: none"> 1. el tutor pasa al modo de edición 2. el sistema muestra la interfase de modo edición 3. El Tutor selecciona el tipo de recurso en la lista desplegable de la caja <i>agregar recurso</i>. 4. El sistema muestra el formulario de <i>Configuración de recurso</i>. 5. El tutor llena el formulario. 6. El tutor selecciona el recurso que se va agregar 7. El sistema valida el recurso. 8. El sistema confirma que el recurso ha sido agregado

Tabla 3.17 Especificación de casos de uso – Ver recurso de módulo o lección

Caso Real:	Ver recursos de la lección o módulo
Actor:	Tutor, Administrador, usuario
Propósito:	Permitir al tutor/administrador ver los recursos que están disponibles para su uso.
Resumen:	<ol style="list-style-type: none"> 1. El usuario ingresa a la sección <i>recursos</i> 2. El sistema muestra todos los recursos disponibles 3. El usuario selecciona el recurso que desea ver 4. El sistema muestra el recurso.

Tabla 3.18 Especificación de casos de uso – Modificar recurso de módulo o lección

Caso Real:	Modificar recursos de la lección o módulo
Actor:	Tutor, Administrador
Propósito:	Permitir al tutor/administrador modificar los parámetros de configuración del recurso que el tutor seleccione.
Resumen:	<ol style="list-style-type: none"> 1. El usuario ingresa a la sección <i>recursos</i> 2. El sistema muestra todos los recursos disponibles 3. El sistema muestra la pantalla de configuración 4. El usuario cambia y modifica los parámetros 5. El tutor guarda los cambios. 6. El sistema hace efectivo las modificaciones.

Tabla 3.19 Especificación de casos de uso – Modificar recurso de módulo o lección

Caso Real:	Modificar recursos de la lección o módulo
Actor:	Tutor, Administrador
Propósito:	Permitir al tutor/administrador modificar los parámetros de configuración del recurso que el tutor seleccione.
Resumen:	<ol style="list-style-type: none"> 1. El usuario ingresa a la sección <i>recursos</i> 2. El sistema muestra todos los recursos disponibles 3. El sistema muestra la pantalla de configuración 4. El usuario cambia y modifica los parámetros 5. El tutor guarda los cambios. 6. El sistema hace efectivo las modificaciones.

Tabla 3.20 Especificación de casos de uso – Eliminar recurso de módulo o lección

Caso Real:	Eliminar recursos de la lección o módulo
Actor:	Tutor, Administrador
Propósito:	Permitir al tutor/administrador eliminar un recurso que el tutor seleccione.
Resumen:	<ol style="list-style-type: none"> 1. el tutor pasa al modo de edición 2. el sistema muestra la interfase de modo edición 3. El usuario selecciona la opción eliminar recurso 4. El sistema muestra el mensaje de advertencia que se va a eliminar un recurso. 5. El usuario confirma o no la eliminación 6. El sistema muestra si el recurso fue eliminado o no

Tabla 3.21 Especificación de casos de uso – Descargar recurso de módulo o lección

Caso Real:	Descargar recursos de la lección o módulo
Actor:	Tutor, Administrador, usuario
Propósito:	Permitir al tutor/administrador descargar el recurso de una lección

Resumen:	<ol style="list-style-type: none">1. El usuario ingresa en el recurso2. el sistema muestra el recurso3. El usuario selecciona guardar recurso en un destino diferente.4. el sistema realiza la operación5. el usuario obtiene el recurso.
----------	---

DESARROLLO DEL CURSO E-LEARNING DE LA MATERIA DE INGENIERÍA DE SOFTWARE I FACILITADO CON EL MÉTODO ERGOGLÍFICO

Para el desarrollo del curso: Ingeniería de Software I, se utilizó el Método ergoglífico el cual contiene las etapas de Análisis, Diseño, Construcción y Evaluación anteriormente mencionadas en el Capítulo I, Sección 1.7.9.2. Estas etapas serán desarrolladas de forma detallada con sus respectivos procedimientos para la implementación del presente curso.

Como referencia debemos acotar que este curso será implantado en la plataforma Moodle, lo que nos permitió usar una gran variedad de recursos que facilitaron el aprendizaje del los educandos.

3.2. ETAPA DE ANÁLISIS

En esta primera etapa vamos analizar los requerimientos que dará lugar a nuestro curso, pero antes debemos tener claro bien los objetivos

Un buen análisis ahorra tiempo y dinero, reduce errores, asigna actividades más precisas y permite seguir un objetivo claro.

En la etapa de Análisis se procede a cumplir las siguientes subfases:

1. Conocimiento de los Usuarios
2. Identificación de las Necesidades de Conocimiento
3. Establecimiento de Objetivos
4. Elaboración del Diagrama de Análisis

Estas subfases se irán detallando en el transcurso del desarrollo del presente curso. Ver Figura 3.7

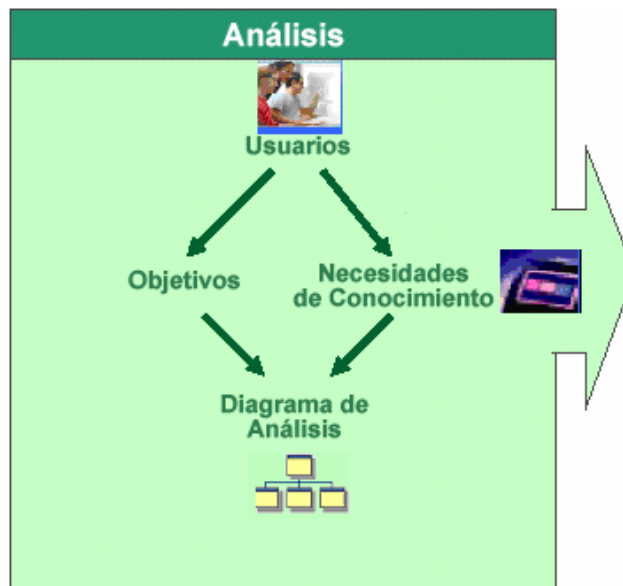


Figura 3.7. Fase de Análisis. ^{3.1}

3.2.1. Conocimiento de los usuarios

Anteriormente se definió a la población potencial que iba ser dirigido el curso. En nuestro caso este curso esta dirigido especialmente a usuarios del Departamento de Ciencias de Computación que toman la materia de "Ingeniería de Software I", que por motivos de encontrarse en diversas partes del Ecuador no pueden seguir de manera continua esta especialidad.

3.2.2. Identificación de las necesidades de conocimiento

Previamente de construir el curso de conocimiento, se identificó la necesidad de conocimientos por parte de los alumnos, es decir, qué conceptos, habilidades, destrezas, motivaciones que necesitan y cual será el nivel de prioridad. En esta etapa la manera mas fácil de obtener estos resultados es mediante una encuesta que se les realiza a los estudiantes, pero como este curso es dirigido a los usuarios del Departamento de Ciencias de Computación no es necesario aplicar ésta.

La información introducida en el curso esta de acuerdo al pensum académico aprobado por el departamento de Ciencias de la Computación.

^{3.1} Grafico Tomado del Documento: "Developing Knowledge Products". William Horton Consulting.

- Proyectos Ingeniería de Software
- Estimación de Proyectos de Software
- Gestión de proyectos de Software

A partir de estos temas se procedió a recolectar la información referente a: Proyectos de Ingeniería de Software, Estimación de proyectos de Software y Gestión de proyectos de Software para consecutivamente en la etapa de diseño estructurar las unidades y lecciones, establecer y organizar las actividades de enseñanza, ejercicios y tests.

3.2.3. Establecimiento de objetivos

El objetivo general del proyecto es el mismo que resume lo que el curso quiere obtener (producto de conocimiento). Existe la fórmula para identificar el objetivo general que es: Describir el grupo de personas que van a utilizar el producto y lo que van a hacer con este producto, como se puede ver en la Figura 3.8

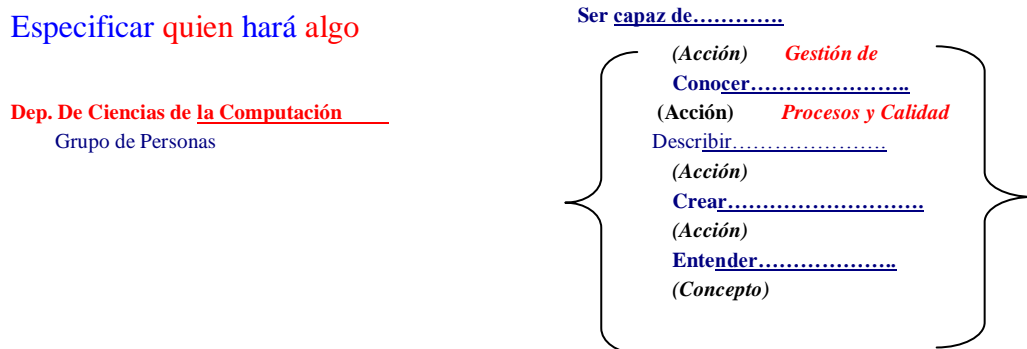


Figura 3.8 Establecimiento del Objetivo General. ^{3.2}

A continuación se procede a establecer los objetivos específicos. Un *objetivo específico se compone de una meta más un criterio que determina como se lo va a realizar.* Ver Figura 3.9

^{3.2} Tomado de la tesis de desarrollo de un Wbt facilitado con el método ergoglíptico perteneciente a Rina Pacheco

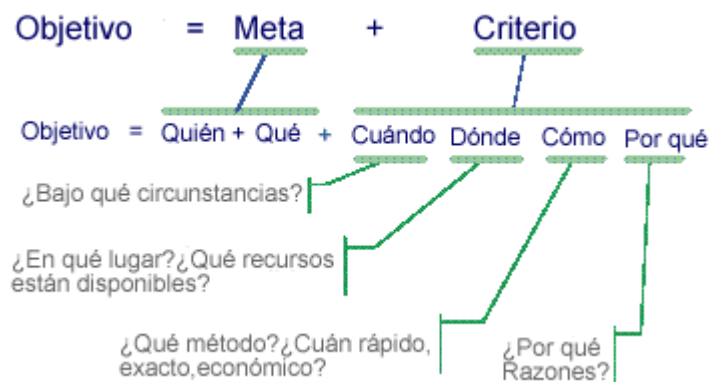


Figura 3.9 Establecimiento de Objetivos Específicos.^{3.3}

El fin específica *quién* y *qué*, es decir, quien va a hacer algo. El criterio en cambio añade cuatro componentes: *cuándo*, *dónde*, *cómo* y *por qué*.

Cuándo. Pregunta bajo que circunstancias el hecho va a ser realizado.

Dónde. Pregunta en qué lugar es realizado y qué recursos están disponibles.

Cómo. Hace referencia al método a través del cual va a ser realizado, cuan rápido, exacto y económico debe ser realizado.

Por qué. Pregunta lo que motivará a la persona a realizarlo.

En extracto un objetivo específico requiere concretar quien debe hacer algo así como también cuándo, dónde, cómo y por qué debe hacerlo.

Para el establecimiento de los objetivos específicos del proyecto se puede utilizar las siguientes listas de verbos, sean de índole educacional o de desempeño respectivamente.

Tabla 3.22. Verbos para el Planteamiento de Objetivos.

Conocimientos: Recordar Información	Comprensión: Interpretar Información poniéndola en sus propias palabras	Aplicación: Usar el conociendo o la generalización en una nueva situación
Organizar	Clasificar	Aplicar

^{3.3} Tomado de la tesis de desarrollo de un Wbt facilitado con el método ergoglífico perteneciente a Rina Pacheco

<p>Definir</p> <p>Duplicar</p> <p>Rotular</p> <p>Enumerar</p> <p>Parear</p> <p>Memorizar</p> <p>Nombrar</p> <p>Ordenar</p> <p>Reconocer</p> <p>Relacionar</p> <p>Recordar</p> <p>Repetir</p> <p>Reproducir</p>	<p>Describir</p> <p>Discutir</p> <p>Explicar</p> <p>Expresar</p> <p>Identificar</p> <p>Indicar</p> <p>Ubicar</p> <p>Reconocer</p> <p>Reportar</p> <p>Re-enunciar</p> <p>Revisar</p> <p>Seleccionar</p> <p>Ordenar</p> <p>Decidir</p> <p>Traducir</p>	<p>Escoger</p> <p>Demostrar</p> <p>Dramatizar</p> <p>Emplear</p> <p>Ilustrar</p> <p>Interpretar</p> <p>Operar</p> <p>Preparar</p> <p>Practicar</p> <p>Programar</p> <p>Esbozar</p> <p>Solucionar</p> <p>Utilizar</p>
<p>Análisis: Dividir el conocimiento en partes y mostrar relaciones entre ellas.</p>	<p>Síntesis: Juntar o unir, partes o fragmentos de conocimiento para formar un todo y construir relaciones para situaciones nuevas.</p>	<p>Evaluación: Hacer juicios en base a criterios dados.</p>
<p>Analizar</p> <p>Valorar</p> <p>Calcular</p> <p>Categorizar</p> <p>Comparar</p> <p>Contrastar</p> <p>Criticar</p> <p>Diagramar</p> <p>Diferenciar</p> <p>Discriminar</p> <p>Distinguir</p> <p>Examinar</p>	<p>Organizar</p> <p>Ensamblar</p> <p>Recopilar</p> <p>Componer</p> <p>Construir</p> <p>Crear</p> <p>Diseñar</p> <p>Formular</p> <p>Administrar</p> <p>Organizar</p> <p>Planear</p> <p>Preparar</p>	<p>Valorar</p> <p>Argumentar</p> <p>Evaluar</p> <p>Atacar</p> <p>Elegir</p> <p>Comparar</p> <p>Defender</p> <p>Estimar</p> <p>Evaluar</p> <p>Juzgar</p> <p>Predecir</p> <p>Calificar</p>

<p>Experimentar</p> <p>Inventariar</p> <p>Cuestionar</p> <p>Examinar</p>	<p>Proponer</p> <p>Trazar</p> <p>Sintetizar</p> <p>Redactar</p>	<p>Otorgar puntaje</p> <p>Seleccionar</p> <p>Apoyar</p> <p>Valorar</p>
--	---	--

La formulación de los objetivos del curso no es una actividad fácil, al contrario esto lleva un proceso de tiempo lo que permitirá de manera confiable obtener estos. Los objetivos específicos del curso e-learning se muestran en el formulario CEL-IS-OBJETIVOS-01.

Tabla 3.23 OBJETIVOS

NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software		
IMPLEMENTACIÓN	Departamento de Ciencias de Computación		
MATERIA	Ingeniería de Software I		
CREADORES	Alejandro Carrera, Gabriela del castillo		
DIRIGIDO A	Alumnos del Séptimo nivel Personal Militar de las Fuerzas Armadas		
OBJETIVOS	Desarrollar proyectos de Ingeniería de Software		
APLICACIÓN	Emplear en las Organización o Empresas del Ecuador		
OBJETIVOS			
QUIEN?	Dirigido a:		
	Estudiantes , personal militar quienes necesitan aprender ingeniería de software	Estudiantes , personal militar quienes necesitan aprender ingeniería de software	Estudiantes , personal militar quienes necesitan aprender Gestión de proyectos de Software
QUE?	Objetivos de instrucción		
	<input checked="" type="checkbox"/> Definir <input checked="" type="checkbox"/> Analizar <input checked="" type="checkbox"/> Describir <input type="checkbox"/> Organizar <input type="checkbox"/> Solucionar <input type="checkbox"/> Evaluar	<input type="checkbox"/> Definir <input checked="" type="checkbox"/> Analizar <input checked="" type="checkbox"/> Describir <input checked="" type="checkbox"/> Organizar <input checked="" type="checkbox"/> Solucionar <input type="checkbox"/> Evaluar	<input checked="" type="checkbox"/> Definir <input checked="" type="checkbox"/> Analizar <input checked="" type="checkbox"/> Describir <input checked="" type="checkbox"/> Organizar <input checked="" type="checkbox"/> Solucionar <input checked="" type="checkbox"/> Evaluar
CUANDO?	Tiempo		
	Después de revisar y estudiar la sección de definición de Ingeniería de Software	Después de revisar y estudiar la sección de estimación de Proyectos	Después de revisar y estudiar la sección de estimación de Proyectos
DONDE?	Lugar		
	En su hogar	En su oficina	En su hogar
COMO?	Método		
	Accediendo a la información correspondiente a Introducción a la ingeniería de Software	Accediendo a la información correspondiente estimación de proyectos	Accediendo a la información correspondiente a Gestión de Proyectos
POR QUE?	Estimulo		
	Por que permite un mejor desarrollo de conocimientos de valores y aptitudes	Por que permite un mejor desarrollo de conocimientos de valores y aptitudes	Porque están realizadas en base a una metodología instruccional

3.2.4. Elaboración del diagrama de análisis

En esta etapa realizamos un diagrama de análisis del proyecto y es donde obtenemos toda la información necesaria para llegar a compromisos, acuerdos, decisiones que se tomaron en la etapa de análisis, y que ayudará como referente para el fiel cumplimiento de las demás etapas del desarrollo del curso e-learning CEL-IS-PROYECTO-01.

Tabla 3.24 Diagrama de análisis

PROYECTO			
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software		
IMPLEMENTACIÓN	Departamento de Ciencias de Computación		
MATERIA	Ingeniería de Software I		
CREADORES	Alejandro Carrera, Gabriela del castillo		
USUARIO	E.S.P.E Av. Del Progreso s/n		
EJECUCIÓN	Sangolquí/ESPE 05/01/2007 E-learning (Moodle)		
OBJETIVOS DEL PROYECTO	Incrementar la capacidad de autoaprendizaje e investigación por parte de los estudiantes y personal militar en el área de Ingeniería , Software		
DISEÑADORES	Alejandro Carrera Gabriela de castillo		
DIRECTOR DEL PROYECTO	_____ Ing. Edgar Hermosa		
Personal de Producción	<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; vertical-align: top;"> Contenido: Ing. Tatiana Gualotuña Alejandro Carrera Gabriela del Castillo </td> <td style="width: 50%; vertical-align: top;"> Ilustración: Gabriela del Castillo Animación: Alejandro Carrera Audio Y Video: Alejandro Carrera </td> </tr> </table>	Contenido: Ing. Tatiana Gualotuña Alejandro Carrera Gabriela del Castillo	Ilustración: Gabriela del Castillo Animación: Alejandro Carrera Audio Y Video: Alejandro Carrera
Contenido: Ing. Tatiana Gualotuña Alejandro Carrera Gabriela del Castillo	Ilustración: Gabriela del Castillo Animación: Alejandro Carrera Audio Y Video: Alejandro Carrera		
Recursos	Manuales: Ingeniería de Software, Macromedia Flash 2004, Moodle Video y Audio: Camtasia, Loquendo,		
Experto en la Materia	_____ Ing. Tatiana Gualotuña		
Personal de Desarrollo	<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; vertical-align: top;"> Herramientas de Autor: Alejandro Carrera Gabriela del Castillo </td> <td style="width: 50%; vertical-align: top;"> Programación: Alejandro Carrera Ensamblaje: Alejandro Carrera Gabriela del castillo </td> </tr> </table>	Herramientas de Autor: Alejandro Carrera Gabriela del Castillo	Programación: Alejandro Carrera Ensamblaje: Alejandro Carrera Gabriela del castillo
Herramientas de Autor: Alejandro Carrera Gabriela del Castillo	Programación: Alejandro Carrera Ensamblaje: Alejandro Carrera Gabriela del castillo		

3.3. ETAPA DE DISEÑO

El diseño es el segundo paso del desarrollo del curso e-learning, en esta fase se utiliza la información obtenida en la fase del análisis para determinar el contenido del curso.

Partiendo de los objetivos del curso se determinará una estrategia de conocimiento que acompañará a estos objetivos.

Una vez elegida la estrategia de conocimiento continuamos con la labor de diseñar los módulos del contenido. Iniciando con un módulo de alto nivel, el mismo que se descompondrá en módulos sub-módulos (Como se muestra en la Figura 3.10.) los cuales están compuestos de objetos de conocimiento simples.

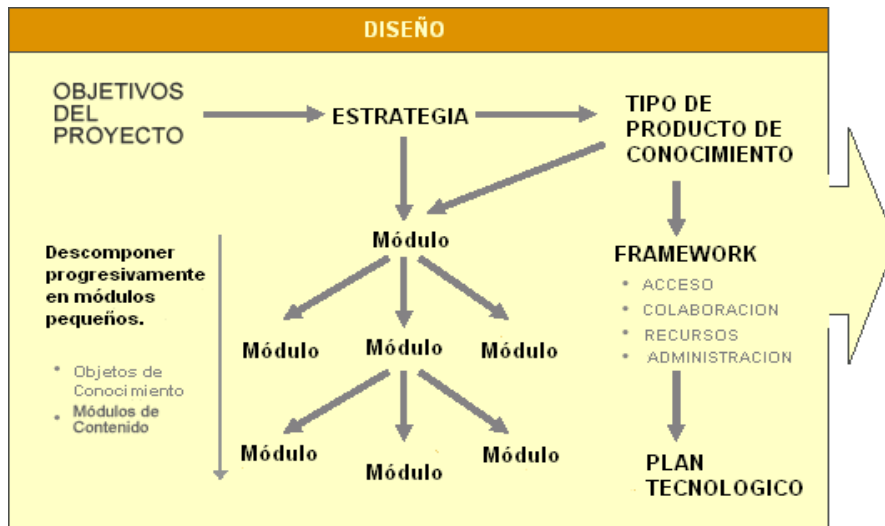


Figura 3.10. Diagrama de Diseño.^{3,4}

El frameWork del curso: Se compone de mecanismos de acceso como: menú, índice, enlaces, mapa; mecanismos de colaboración como: e-mail, chats, foros, consultas; recursos relacionados como: glosario y herramientas de administración como registro de usuarios, seguridad.

Recordemos que Moodle (Modular Object Oriented Distance Learning Enviroment).es un paquete de software para la creación de cursos y sitios Web basados en Internet. Es un proyecto en desarrollo diseñado para dar soporte a un marco de educar social contractiva. Por tal motivo el frameWork del curso es completamente de Moodle.

En esta etapa se procederá a realizar estas subfases:

- Elegir la estrategia
- Seleccionar el tipo de producto de conocimiento
- Definir los objetos de conocimiento
- Diseño preciso del contenido
- Determinación del FrameWork
- Lista de requerimientos Tecnológicos

Como podemos describir en síntesis estas subfases se irán especificando en el desarrollo del presente curso E-Learning tomando en consideración una unidad como

^{3,4} Tomado del Documento: "Developing Knowledge Products". William Horton Consulting.

referencia así como también su contenido (lección) específico, como son: La unidad de Proyectos Ingeniería de Software y la lección Conceptos y definición de Ingeniería de Software. Los formularios serán llenados de acorde a la misma unidad y lección.

3.3.1. SELECCIONAR LA ESTRATEGIA

La estrategia es el acercamiento general para alcanzar los objetivos del curso. A continuación señalaremos lo siguiente:

3.3.1.1. Estrategias Populares de Conocimiento

Las estrategias principales usadas para los cursos son: son *Informativa*, *Conductiva*, *Cognoscitiva* y *Constructivista*.

La estrategia informativa.- ayuda a los usuarios a buscar información cuando ellos más lo necesitan. Los usuarios aprendices poseen muy poco conocimiento lo que esto es previo transferible directamente o habilidades acerca de los contenidos. En esta etapa de diseño clásico es el más adecuado por que ya esta determinado, es secuencial y se utilizan referencias, lo que nos permitirá a los estudiantes desenvolver sus propios conocimientos que les sirvan como referencia para futuras experiencias.

La estrategia conductiva.- refuerza las conductas deseadas, esta se basa en cambios observados en la conducta del sujeto, además esta se enfoca hacia la repetición de patrones de conducta hasta que estos se puedan realizar de manera automática, cabe anotar que esto se toma un poco de tiempo. Las actividades que requieren bajo procesamiento (memorización) podemos facilitarlo mediante estrategias mas frecuentemente relacionadas con salidas conductivas (reforzamiento).

La estrategia cognoscitiva.- se basa sobre la investigación en la psicología del aprendizaje para diseñar información que se transfiera a la memoria a largo plazo. Esta consiste en los procesos que tienen lugar atrás de los cambios de conducta. Estos cambios son observados para usarse como indicadores para entender lo que esta pasando en la mente del que aprende. Las tareas que requieren un nivel superior de procesamiento (por ejemplo, clasificación, reglas o ejecuciones de procedimientos)

están principalmente asociadas con las estrategias que tienen un fuerte énfasis cognoscitivista.

Estrategia Constructivista.- es la de ayuda a los aprendices a construir su propio conocimiento por el descubrimiento y la exploración ya sea individual o grupal. Se sustenta en la premisa de que cada persona construye su propia perspectiva del mundo que le rodea a través de sus propias experiencias y esquemas mentales desarrollados. El constructivismo se enfoca en la preparación del que aprende para resolver problemas en condiciones ambiguas. Las tareas que demandan altos niveles de procesamiento (tales como: solución de problemas heurísticos (invención), selección de personal y monitoreo de estrategias cognitivas) frecuentemente son aprendidas mejor con estrategias avanzadas desde una perspectiva constructivista.

Tabla 3.25 fortalezas y debilidades ^{3.5}

	DEBILIDADES	FORTALEZAS
Conductismo	<ul style="list-style-type: none"> • El que aprende podría encontrarse en una situación en la que el estímulo para la respuesta correcta nunca ocurre, por lo tanto el aprendiz no responde. • Un trabajador al que se le ha condicionado solo para responder a ciertas situaciones de problemas en el lugar de trabajar, de pronto puede detener la producción cuando sucede algo anormal y el no es capaz de encontrar una solución por no entender el sistema. 	<ul style="list-style-type: none"> • El que aprende sólo tiene que concentrarse en metas claras y es capaz de responder con rapidez y automáticamente cuando se le presenta una situación relacionada con esas metas. Fueron condicionados para reaccionar a las siluetas de los aviones enemigos, la respuesta que se esperaría sería automática.
Cognoscitivismo	<ul style="list-style-type: none"> • El aprendiz aprende a realizar una tarea, pero podría no ser la mejor forma de realizarla o la más adecuada para el aprendiz o la situación. Por ejemplo, acceder al Internet en una computadora podría no ser lo mismo que acceder en otra computadora. 	<ul style="list-style-type: none"> • la meta es capacitar al aprendiz para que realice tareas repetitivas y que aseguren consistencia. Acceder dentro y fuera a una computadora del trabajo es igual para todos los empleados; es importante realizar la rutina para evitar problemas.
Constructivismo	<ul style="list-style-type: none"> • En una situación donde la conformidad es esencial, el pensamiento divergente y la iniciativa podrían ser un problema. Tan solo imaginemos, lo que sucedería con los fondos fiscales, si todos decidiéramos pagar impuestos de acuerdo a los criterios de cada quien 	<ul style="list-style-type: none"> • como el que aprende es capaz de interpretar múltiples realidades, está mejor preparado para enfrentar situaciones de la vida real.

Una vez establecido las estrategias existentes se escogió la estrategia constructivista formal por que:

- Los usuarios pueden edificar el conocimiento a partir de sus experiencias.
- Los usuarios son garantes de su aprendizaje

^{3.5} Schuman, 1996

- Los usuarios escogen su propio equilibrio de aprendizaje
- Los usuarios aplican conocimientos en actividades del mundo real.
- Los usuarios eligen el curso que desea aprender
- Que el usuario encuentre lo que busca en el curso
- Los usuarios pueden manejar los métodos y herramientas que concierten adecuadamente su aprendizaje

3.3.2. SELECCIONAR EL TIPO DE PRODUCTO DE CONOCIMIENTO

Para seleccionar el tipo de producto de conocimiento, se debe tomar en cuenta tres características que diferencian los tipos de productos de conocimiento.

3.3.2.1. Líder – Producto vs. Líder – Consumidor

Cuando el creador tiene el control de contenido del curso, el usuario es completamente neutral, por lo que es un receptor no tiene ninguna oportunidad de envolver sus inquietudes a menos que el usuario tome por su propia cuenta puesto que los curso puede ser elegidos sin ningún Tutor.

En cambio cuando el usuario es dirigente de su aprendizaje, el usuario es el que toma decisiones en el curso, como ejemplo que actividades va a realizar, cuando salir, cunado entrar, que horario, en si el toma esta decisión.

3.3.2.2. Asíncrono vs. Sincrono

Síncrono: Todos los implicados en una actividad deben realizar su tarea al mismo tiempo o sea en tiempo real.

Asíncrono: Las actividades de los participantes pueden experimentarse en el momento que lo deseen.

Los productos de conocimiento no son puramente síncronos, o asíncronos pueden ser una combinación de las características de los dos, ya que en esto existe tareas que pueden tomar un tiempo para ser entendida de manera correcta.

3.3.2.3. Tamaño de la clase

Estos pueden ser grupos grandes o pequeños, pero se debe tener en cuenta las diferentes características que pueden tener estos grupos, por ejemplo si es un grupo pequeño, este puede ser personalizado, independiente, por el contrario si el grupo es grande este tomara mas tiempo y será mas complejo de manejarlo, en consecuencia hay que tener mucho aplicación al momento de escoger el tamaño.

3.3.2.4. Escoger el tipo de Curso

Características del Curso E-Learning:

- Dirigido a usuarios del Departamento de Ciencias de la Computación.
- Estudiantes que necesiten mejorar sus conocimientos en Ingeniería de Software.
- Estudiantes que tengan una base de la materia de ingeniería de software.

Estas características son las bases para la elaboración de este curso E-Learning debido a que el estudiante podrá desarrollar su capacidad de autoaprendizaje, creatividad, claro esta teniendo un facilitador como es el tutor quien le guiara durante todo el proceso del curso.

3.3.3. ESPECIFICACIÓN DE LOS OBJETOS DE CONOCIMIENTOS

3.3.3.1. Definición del Objeto de conocimiento

Un objeto de conocimiento es un fragmento del contenido del curso y también pueden ser páginas Web, ayudas, video, sonido, etc. Cada objeto de conocimiento se crea en base a un objetivo. Los objetos de conocimiento pueden ser de 3 tipos:

- Objetos de conocimiento de nivel Superior. (Curso)
- Objetos de conocimiento de nivel Medio.(Unidades y Lecciones)
- Objetos de conocimiento de nivel Inferior. (Páginas)

El establecimiento de los objetos de conocimiento del presente curso “Ingeniería de Software I” se realizo con la ayuda de la Ing. Tatiana Gualotuña en base sus

experiencias como docente en el área. Para la estructuración del contenido del curso se precedió a dividir la información en 3 unidades, que constan con sus respectivas lecciones además se estableció actividades autodidácticas, test, retroalimentación, elementos multimedia como son graficas, animaciones y videos.

3.3.3.2.Contenido de un objeto de Conocimiento

Un objeto de conocimiento de nivel superior esta constituido en sub-módulos secuenciales. Y en caso de que sea objeto de conocimiento de nivel inferior se determina previamente la experiencia de aprendizaje específica.

Además del contenido, el objeto de conocimiento requiere de otros módulos tales como:

- Plan de la Lección
- Bienvenida
- Introducción
- Técnicas de Motivación
- Actividades de Aprendizaje
- Resumen
- Tests o Prácticas
- Evaluaciones
- Mecanismos de Acceso

3.3.3.3.Cuerpo de una Página (Objeto de Conocimiento de Nivel Inferior)

1. Objeto de conocimiento de nivel inferior ó una página.
2. El título mostrado como un banner o una cabecera ubicado al inicio de la página.
3. Botones referentes a las unidades con los que cuenta el curso.
4. El enfoque principal de la página debe ser el contenido actual que debe ubicarse en la parte derecha. Este contenido puede ser texto, gráficos y multimedia.

5. En la parte inferior debe ubicarse links a otras páginas de conocimiento de nivel medio, así como actividades, prácticas y el resumen.

3.3.3.4. Cuerpo de las lecciones (Objeto de conocimiento de nivel medio)

Una lección está formada por varias páginas u objetos de conocimiento de nivel inferior.

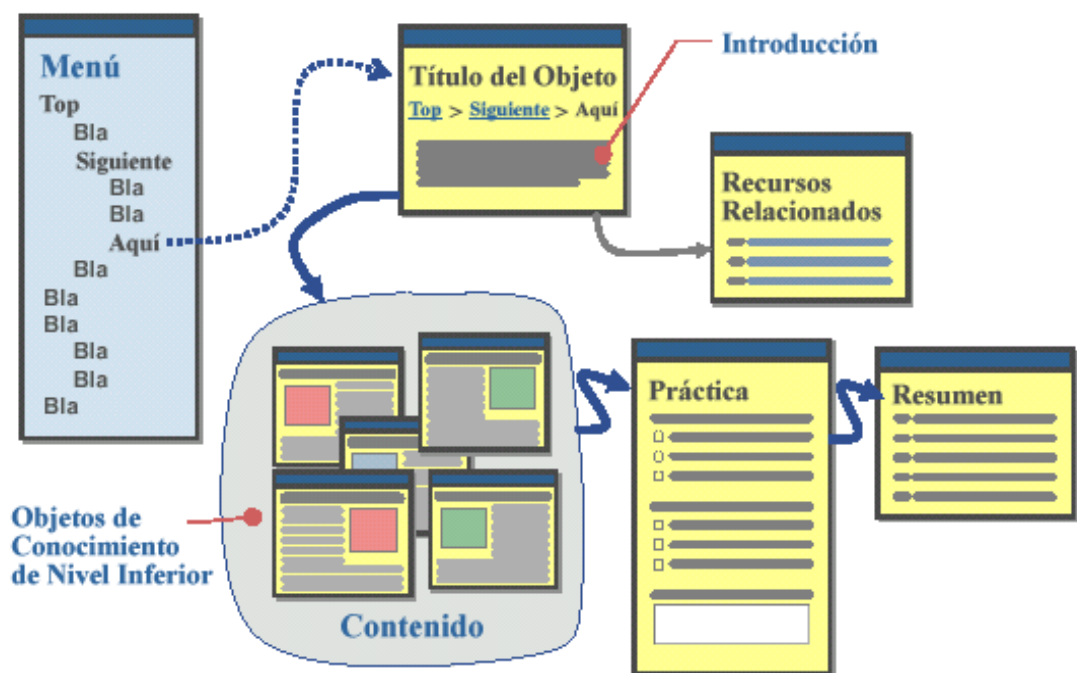


Figura.3.11 Lecciones^{3.6}

Recordemos que para el desarrollo del curso E-learning de la materia de Ingeniería de Software I y su implementación vamos a utilizar la plataforma de educación virtual Moodle. En moodle nos da la opción de escoger diferentes maneras de llevar nuestro curso, en nuestro caso hemos escogido el modo de lecciones (Temas) ya que nuestro curso es amplio y complejo.

El usuario para ver una lección previamente debe haber seguido unos pasos para ser considerado usuario o alumno del curso. Las lecciones del curso van a tener un contenido similar durante todo el curso, esto dependerá de lo complejo que sea la unidad o lección.

^{3.6} Grafico Tomado del Documento: “Developing Knowledge Products”. William Horton Consulting.

Después de haber ingresado al aula virtual el usuario tendrá una vista inicial de los cursos disponibles que dispone la Escuela Politécnica del Ejército, seguidamente elegirá el curso de Ingeniería de Software I.

Una vez ingresado el usuario tendrá una vista amplia de las opciones que posee la plataforma, seguido en la parte central del inicio del curso y las lecciones que contiene la materia de ingeniería de software. Recordemos que los usuarios deben seguir una secuencia adecuada que el tutor crea conveniente.

A continuación de experimentar el inicio del curso, el usuario puede dirigirse a una presentación como la bienvenida, introducción, actividades de aprendizaje, resumen, etc. Finalmente el usuario terminará la lección con una reflexión o mensaje.

3.3.3.5. Jerarquía de objetos de conocimiento

Lo primero que se hace en el diseño de objetos de conocimiento es crear un esquema de la distribución del curso en base al objetivo general y específicos del curso. La siguiente estructuración pertenece al curso “*Ingeniería de Software I*” (Ver Figura 3.12).

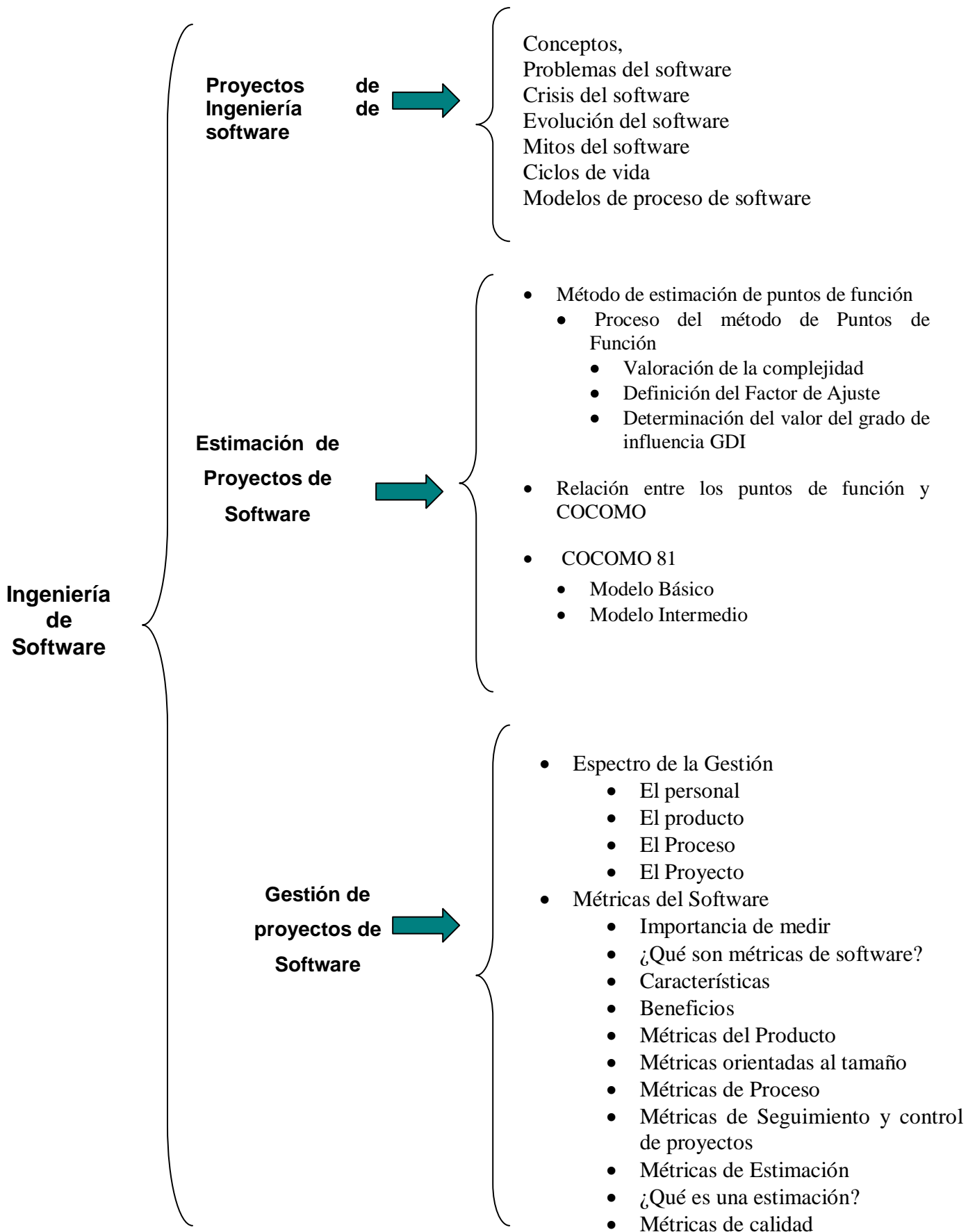
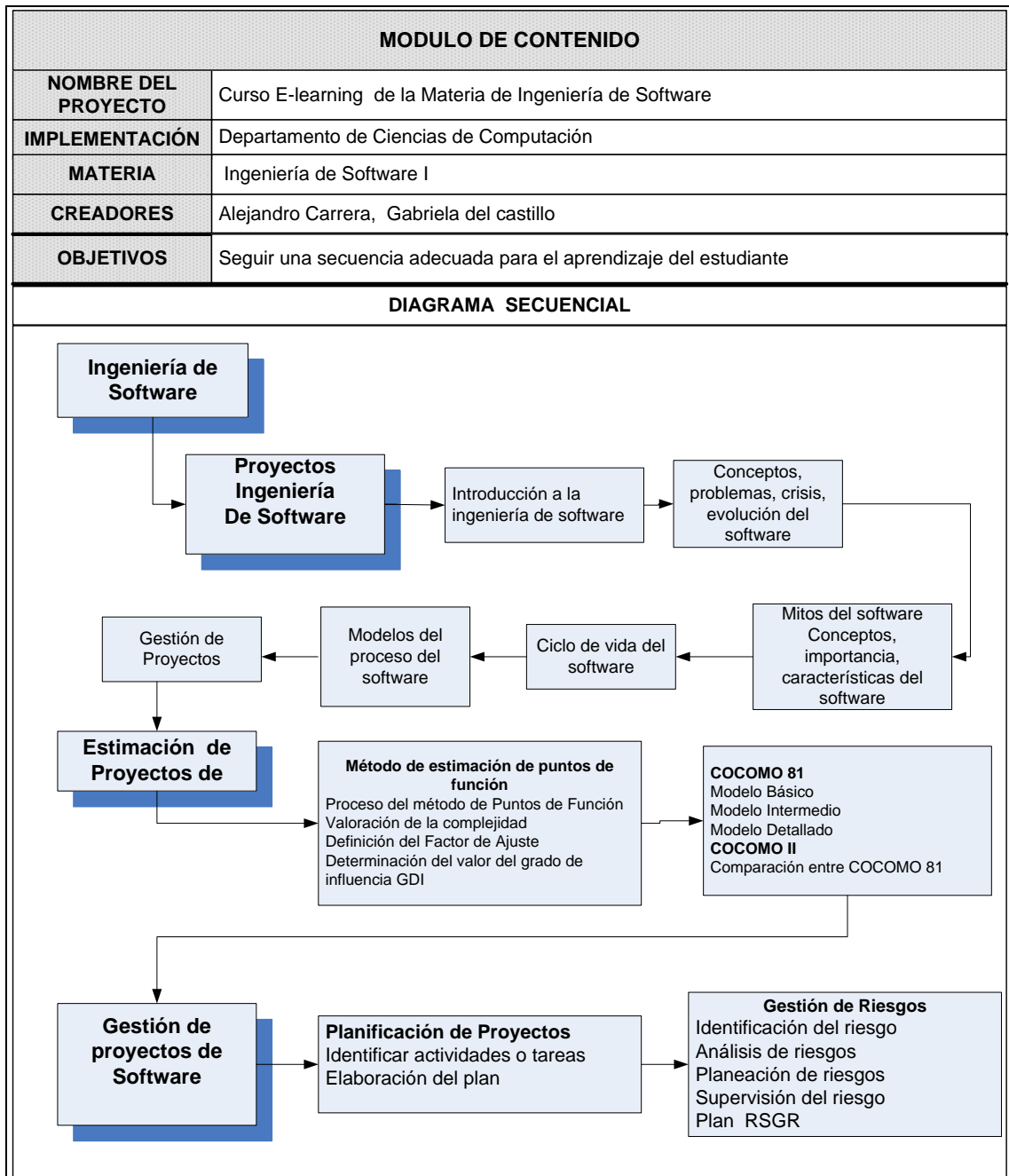


Figura 3.12 Jerarquía de Objetos de Conocimiento.

3.3.3.6. Diagrama del Contenido del Curso

Muestra como esta organizado el curso, de los objetos a sub.-menús de un producto de conocimiento. Este diagrama muestra como un mapa de navegación el cual el usuario puede ver como esta conformado el curso, para el diseño de este mapa se utilizo el siguiente formulario CEL-IS-MODULO _ CONTENIDO-01.

Tabla 3.26 Diagrama del contenido del curso



3.3.3.7. Objeto de Conocimiento de Nivel Superior

El objeto de conocimiento de nivel superior simboliza el contenido del curso. Para el diseño de este objeto de conocimiento se utilizó el siguiente formulario. Ver Formulario CEL-IS-OCNS-01.

Tabla 3.27 Objeto de conocimiento de nivel superior

CONOCIMIENTO DE NIVEL SUPERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del castillo
OBJETIVOS	Aprendizaje de la materia de Ingeniería de Software
SALIDAS	Slide (flash), WEB (HTML)
DESCRIPCION	
TITULO	Ingeniería de Software
INTRODUCCION	bienvenida_e-learning (.swf)
CONTENIDO	<ul style="list-style-type: none"> ■ Bienvenida Archivo: CEL-IS-BIENVENIDA-01 ■ Plan de Leccion Archivo: CEL-IS-PLECCION-01 ■ Importancia Archivo: CEL-IS-IMPORTANCIA-01 ■ Contenido Archivo: CEL-IS-CONTENIDO-01
Actividades	Ninguno
DESCRIPCION	Contiene la información para que el estudiantes se entere de la materia, las actividades del curso, contenido, las lecciones y de todo el materia apara que pueda aprender.

3.3.3.8. Objeto de Conocimiento de nivel medio

Describe las unidades y lecciones del curso. Para el diseño de este objeto de conocimiento se utilizó los siguientes formularios. Ver Formulario CEL-IS-OCNM-02 (Unidad) y Formulario CEL-IS-OCNM-03 (Lección).

Tabla 3.28 Objeto de Conocimiento de nivel medio

CONOCIMIENTO DE NIVEL MEDIO					
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software				
IMPLEMENTACIÓN	Departamento de Ciencias de Computación				
MATERIA	Ingeniería de Software I				
CREADORES	Alejandro Carrera, Gabriela del castillo				
OBJETIVOS	Los estudiantes aprenderán y entenderán la materia de Ingeniería de software				
SALIDAS	Slide (flash), WEB (HTML)				
MONUMENCLATURA	CEL-IS-NOMBRE-NUMERO				
DESCRIPCION					
TITULO	Introducción a la Ingeniería de Software				
INTRODUCCION	El proyecto de software implica una metodología de desarrollo así como la utilización de técnicas, herramientas para aplicar procedimientos a ser utilizado en el proyecto informático.				
CONTENIDO	<table border="0"> <tr> <td>■ Lecciones Archivo: CEL-IS-NOMBRE-01</td> <td>■ LECCIÓN XX Archivo: CEL-IS-LECCION-XX</td> </tr> <tr> <td>■ Actividades Archivo: CEL-IS-ACTIVIDADES-XX</td> <td>■ LECCIÓN XX Archivo: CEL-IS-LECCION-XX</td> </tr> </table>	■ Lecciones Archivo: CEL-IS-NOMBRE-01	■ LECCIÓN XX Archivo: CEL-IS-LECCION-XX	■ Actividades Archivo: CEL-IS-ACTIVIDADES-XX	■ LECCIÓN XX Archivo: CEL-IS-LECCION-XX
■ Lecciones Archivo: CEL-IS-NOMBRE-01	■ LECCIÓN XX Archivo: CEL-IS-LECCION-XX				
■ Actividades Archivo: CEL-IS-ACTIVIDADES-XX	■ LECCIÓN XX Archivo: CEL-IS-LECCION-XX				
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexion				
DESCRIPCION	Este provee la suficiente información para que el estudiante desarrolle, investigue, aprenda y por ultimo aplique los conocimiento adquiridos por el material dado en el sitio				

3.3.3.9. Objeto de Conocimiento de nivel Inferior

Describe las páginas que conforman la lección del curso. Para el diseño de este objeto de conocimiento se utilizó el siguiente formulario. Ver Formularios desde CEL-IS-OCNI-00 hasta CEL-IS-OCNI- 12.

Tabla 3.29 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del castillo
OBJETIVOS	Dar un pensamiento de reflexión para motivar al alumno antes de iniciar la lección
SALIDAS	Slide (flash), WEB (HTML)
MONUMENCLATURA	CEL-IS-BIENVENIDA-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html
DESCRIPCIÓN	Contenido de la reflexión del curso o lección
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ Lección Archivo: CEL-IS-BIENVENIDA-01 Bienvenida a la lección XX
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexion
DESCRIPCION	Este provee la suficiente información para que el estudiante obtenga la suficiente motivación para iniciar el curso, poniendo en manifiesto de que el obtendrá la suficiente ayuda para iniciar y desarrollar la lección

Tabla 3.30 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del castillo
OBJETIVOS	Dar un pensamiento de reflexión para mostrar al alumno el contenido de la lección
SALIDAS	Slide (flash), WEB (HTML)
MONUMENCLATURA	CEL-IS-BIENVENIDA-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html
DESCRIPCIÓN	Título, contenido
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ Lección Archivo: CEL-IS-BIENVENIDA-01 Introducción
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexion
DESCRIPCION	Este provee información para que el estudiante tenga una noción o descripción de la lección a estudiar

Tabla 3.31 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del castillo
OBJETIVOS	Dar al estudiante los objetivos de la lección
SALIDAS	Slide (flash), WEB (HTML)
MONUMENCLATURA	CEL-IS-BIENVENIDA-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html
DESCRIPCIÓN	Título, contenido
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ Lección Archivo: CEL-IS-BIENVENIDA-01 Objetivos
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexion
DESCRIPCION	Este provee información para que el estudiante tenga una noción o descripción de las diferentes destrezas que adquirirá

Tabla 3.32 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del castillo
OBJETIVOS	Dar al estudiante los temas que se trataran en la lección
SALIDAS	Slide (flash), WEB (HTML)
MONUMENCLATURA	CEL-IS-BIENVENIDA-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html
DESCRIPCIÓN	Titulo, contenido
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ Lección Archivo: CEL-IS-BIENVENIDA-01 Contenido
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexion
DESCRIPCION	Este provee información para que el estudiante tenga una noción o descripción de los temas a estudiar en la lección

Tabla 3.33 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del castillo
OBJETIVOS	Mostrar al estudiante las diferentes actividades a realizar en la lección
SALIDAS	Slide (flash), WEB (HTML)
MONUMENCLATURA	CEL-IS-BIENVENIDA-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html, pdf
DESCRIPCIÓN	Título, contenido
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ Lección Archivo: CEL-IS-BIENVENIDA-01 Actividades a realizar
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexion
DESCRIPCION	Este provee información para que el estudiante tenga la información necesaria para realizar las actividades didácticas como foros, asesorías, etc.

Tabla 3.34 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del castillo
OBJETIVOS	Describir al estudiantes los diferentes materiales a utilizar para el desarrollo del curso
SALIDAS	Slide (flash), WEB (HTML)
MONUMENCLATURA	CEL-IS-BIENVENIDA-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html, pdf
DESCRIPCIÓN	Titulo, contenido
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ Lección Archivo: CEL-IS-BIENVENIDA-01 Recursos
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexion
DESCRIPCION	Este provee información para que el estudiante tenga los materiales a utilizar en la lección como por ejemplo libros, diapositivas, etc.

Tabla 3.35 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del castillo
OBJETIVOS	Enumerar las actividades de evaluación
SALIDAS	Slide (flash), WEB (HTML), hotpotatoes
MONUMENCLATURA	CEL-IS-BIENVENIDA-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html, pdf
DESCRIPCIÓN	Título, contenido
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ Lección Archivo: CEL-IS-BIENVENIDA-01 Evaluación y retroalimentación
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexion
DESCRIPCION	Este provee información para que el estudiante tenga los referencia de las actividades de evaluación, retroalimentación, asesorías, exámenes, pruebas, etc., que se realizan en cada lección

Tabla 3.36 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera
OBJETIVOS	Describir los objetivos específicos de la lección
SALIDAS	Slide (flash), WEB (HTML)
MONUMENCLATURA	CEL-IS-INTRODUCCION-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html, pdf
DESCRIPCIÓN	Título, contenido
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ TITULO Archivo: CEL-IS-INTRODUCCIÓN-01 Objetivos específicos
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexión
DESCRIPCIÓN	Este provee información para que el estudiante tenga en claro los objetivos específicos de la lección

Tabla 3.37 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del castillo
OBJETIVOS	Describir los requerimientos de la lección
SALIDAS	Slide (flash), WEB (HTML)
MONUMENCLATURA	CEL-IS-INTRODUCCION-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html, pdf
DESCRIPCIÓN	Titulo, contenido
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ TITULO Archivo: CEL-IS-INTRODUCCIÓN-01 Requerimientos
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexion
DESCRIPCION	Este provee información para que el estudiante ponga en énfasis valores morales, éticos para el desarrollo de sus trabajos y actividades que se requiera por parte del estudiante.

Tabla 3.38 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del castillo
OBJETIVOS	Contiene el desarrollo o contenido de la lección. Material de apoyo de los temas claves.
SALIDAS	Slide (flash), WEB (HTML)
MONUMENCLATURA	CEL-IS-INTRODUCCION-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html, pdf
DESCRIPCIÓN	Título, Temas, Subtemas
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ TITULO Archivo: CEL-IS-INTRODUCCIÓN-01 Contenido-Introducción
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexión
DESCRIPCIÓN	Lograr que el aprendiz conozca los principios, conceptos, los inicios de la ingeniería de software

Tabla 3.39 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del castillo
OBJETIVOS	Contiene detalladamente las actividades a realizar
SALIDAS	Slide (flash), WEB (HTML)
MONUMENCLATURA	CEL-IS-RECURSOS-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html, pdf
DESCRIPCIÓN	Título, Temas, Subtemas
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ TITULO Archivo: CEL-IS-RECURSOS-01 Actividades de Aprendizaje
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexión
DESCRIPCIÓN	Este contiene la accesoria didáctica a realizar por parte del estudiantes. Tareas, foros, talleres.

Tabla 3.40 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del castillo
OBJETIVOS	Describir las conclusiones obtenidas al termino de la lección
SALIDAS	Slide (flash), WEB (HTML)
MONUMENCLATURA	CEL-IS-RESUMEN-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html, pdf
DESCRIPCIÓN	Titulo, Temas, Subtemas
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ TITULO <li style="padding-left: 20px;">Archivo: CEL-IS-RESUMEN-01 <li style="padding-left: 20px;">Resumen
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexion
DESCRIPCION	Este contiene las conclusiones de la lección

Tabla 3.41 Objeto de Conocimiento de nivel Inferior

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera
OBJETIVOS	Incentivar al alumno sobre valores morales y eticos
SALIDAS	Slide (flash), WEB (HTML)
MONUMENCLATURA	CEL-IS-MOTIVACIÓN-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Imagen, texto, gif, html, pdf
DESCRIPCIÓN	Titulo, Temas, Subtemas
DOCUMENTO O PAGINA	<ul style="list-style-type: none"> ■ TITULO Archivo: CEL-IS-MOTIVACIÓN-01 Motivación
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexion
DESCRIPCION	Este contiene mensaje, relatos, vivencias que generen y ayuden a alimentar los valores en los estudiante.

3.3.4. DISEÑO DETALLADO DEL CONTENIDO

Una lección debe estar conformado por:

- **Módulos de Contenido.-** estos deben incluir bloques de texto, gráficas, secuencias de animaciones, clips de vídeo y otras unidades de media para comunicar una idea específica.
- **Test de Preguntas.-** es donde el usuario pone en práctica lo aprendido en una lección.
- **Actividades de Aprendizaje.-** es donde el usuario puede interactuar con el computador, usuario, tutor.

3.3.4.1. Especificación del Módulo de Contenido

Se detalla los elementos de media. Para el diseño del módulo de contenido se tomo como ejemplo la animación correspondiente a la lección I “Evolución del software” Ver Formulario CEL-IS-ANIMACIÓN-01

Tabla 3.42 elementos de media (animación)

CONOCIMIENTO DE NIVEL INFERIOR	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera
OBJETIVOS	Explicar la evolución del software
SALIDAS	Slide (flash)
MONUMENCLATURA	CEL-IS-ANIMACIÓN-01
DESCRIPCIÓN	
FORMATO DE SALIDA	Clip de película, swf,
DESCRIPCIÓN	Texto, descripción, transición, imágenes, 550 x 400
DOCUMENTO O PAGINA	<p>■ TITULO Archivo: CEL-IS-INTRODUCCION-01 Evolución del software</p>
Actividades	Cada lección tiene resumen, retroalimentación, pruebas, reflexión, etc.
DESCRIPCION	<p>Realizado en formato fla, swf e incorporado en el documento de flash</p>

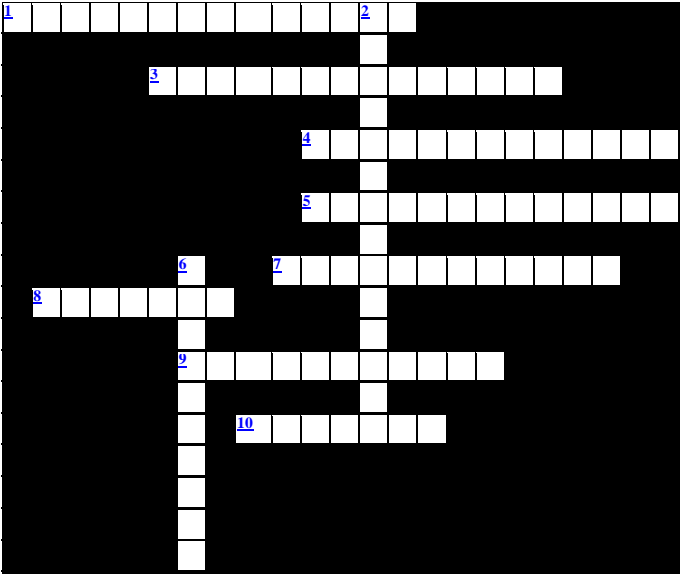
3.3.4.2. Especificación de Test

Existen cuatro tipos de test (completar, ordenar, verdadero/falso, crucigrama), el uno que se encuentra en cada lección del curso y el otro que está en la primera lección. Consiste en el listado de preguntas acerca de la lección actual. Ver Formularios CEL-IS-TEST-01 y CEL-IS-TEST-02.

Tabla 3.43 Especificación de Test

Test de Preguntas	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera, Gabriela del Castillo
OBJETIVOS	Aprendizaje de la materia de Ingeniería de Software
SALIDAS	WEB (HTML)
DESCRIPCIÓN	
Tipo de preguntas	Verdadero o falso & de Selección múltiple
Puntuación	La puntuación dependerá de la ponderación que asigne el tutor en las preguntas
Rastreo	<input checked="" type="checkbox"/> Escoje la respuesta y se guarda automáticamente <input type="checkbox"/> Opción n...n+1 <input type="checkbox"/> Escoje la respuesta y se guarda automáticamente
Objetivo	Medir los conocimientos de la materia de ingeniería de software
Pregunta & Introducción	Selecciones la respuesta correcta en base a sus conocimientos
Respuestas o Retroalimentación	<p>Pregunta de Verdadero o Falso Herramientas.- definen la secuencia en la que se aplican los métodos, las entregas, los controles de calidad y guías para evaluación del progreso</p> <p>Opciones</p> <p><input checked="" type="checkbox"/> Verdadero <input type="checkbox"/> Falso</p> <p>Respuesta y retroalimentación: OK. Las herramientas apoyan a la aplicación de los métodos.. su calificación es: 100% de la preguntas</p> <p>Pregunta de selección múltiple ¿Cual de los siguientes principios pertenecen al Modelo de ciclo en cascada</p> <p><input type="checkbox"/> Define las fases primarias esperadas de ser ejecutadas durante esas fases. <input type="checkbox"/> Provee un espacio de trabajo para la definición de un detallado proceso de desarrollo de software. <input checked="" type="checkbox"/> Definición de los requisitos <input checked="" type="checkbox"/> Implementación y pruebas unitarias</p> <p>Respuesta y retroalimentación: 4 / 4 su puntuación es: 100% de la preguntas</p>

Tabla 3.44 Especificación de Test

Test de Preguntas	
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software
IMPLEMENTACIÓN	Departamento de Ciencias de Computación
MATERIA	Ingeniería de Software I
CREADORES	Alejandro Carrera
OBJETIVOS	Aprendizaje de la materia de Ingeniería de Software
SALIDAS	WEB (HTML). Archivo HotPotatoes
DESCRIPCION	
Tipo de preguntas	Crucigrama del conocimiento
Puntuación	La puntuación dependerá de la ponderación que asigne el tutor al crucigrama
Rastreo	<input checked="" type="checkbox"/> Introducir palabras <input type="checkbox"/> Comprobar <input type="checkbox"/> Pista
Objetivo	Medir los conocimientos de la materia de ingeniería de software mediante test gráficos
Pregunta & Introducción	Llene el crucigrama en base a sus conocimientos
Respuestas o Retroalimentación	 <p>Horizontal:</p> <p>1. Retos: Estos sistemas que incluyen una mezcla de software y hardware son productos que requieren necesariamente la interoperatividad de los dos, ya que estos cumplen funciones específicas complejas. 3. Definen la secuencia en la que se aplican los métodos, las entregas, los controles de calidad y guías para evaluación del progreso.</p> <p>Vertical:</p> <p>2. Mito: Si se está atrasado en el itinerario es posible contratar más programadores para así terminar a tiempo. Realidad: Agregar gente a un proyecto de software atrasado lo atrasa más. 6. Retos: Existe una gran demanda de que exista formalidad en el proceso de desarrollo de software, esto nos garantiza de que el producto o software halla tenidos un transcurso o proceso conocido para se desarrollo.</p>

3.3.4.3. Especificación de las Actividades de Aprendizaje

1. Consiste en especificar las actividades de aprendizaje propuestas dentro de la lección: “DETERMINACIÓN Y CONCEPTUALIZACIÓN DE LOS, MITOS DEL SOFTWARE Y IMPORTANCIA COMO PARTE DE UN PROCESO DE EVOLUCIÓN DE LA INGENIERÍA DE SOFTWARE.”. Ver Formularios CEL-IS-ACTIAPRE-01, CEL-IS-ACTIAPRE-02 CEL-IS-ACTIAPRE-03. Tabla 3.45, Tabla 3.46, Tabla 3.47

Tabla 3.45 Actividad de Aprendizaje

Diagrama de Actividad de Aprendizaje															
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software														
IMPLEMENTACIÓN	Departamento de Ciencias de Computación														
MATERIA	Ingeniería de Software I														
CREADORES	Alejandro Carrera, Gabriela del Castillo														
OBJETIVOS	Aprendizaje de la materia de Ingeniería de Software														
SALIDAS	WORD, PDF														
DESCRIPCION															
Tipo de Actividad	<table border="0"> <tr> <td>Entrenamiento</td> <td>Realizado por</td> </tr> <tr> <td><input type="checkbox"/> Sincrono</td> <td><input type="checkbox"/> Individual</td> </tr> <tr> <td><input checked="" type="checkbox"/> Asincrono</td> <td><input checked="" type="checkbox"/> Equipo de trabajo</td> </tr> <tr> <td>Evaluado por / Retroalimentado</td> <td></td> </tr> <tr> <td><input checked="" type="checkbox"/> Tutor</td> <td><input type="checkbox"/> Computador</td> </tr> <tr> <td><input type="checkbox"/> Otros</td> <td><input checked="" type="checkbox"/> Foros</td> </tr> <tr> <td></td> <td><input checked="" type="checkbox"/> Otros</td> </tr> </table>	Entrenamiento	Realizado por	<input type="checkbox"/> Sincrono	<input type="checkbox"/> Individual	<input checked="" type="checkbox"/> Asincrono	<input checked="" type="checkbox"/> Equipo de trabajo	Evaluado por / Retroalimentado		<input checked="" type="checkbox"/> Tutor	<input type="checkbox"/> Computador	<input type="checkbox"/> Otros	<input checked="" type="checkbox"/> Foros		<input checked="" type="checkbox"/> Otros
Entrenamiento	Realizado por														
<input type="checkbox"/> Sincrono	<input type="checkbox"/> Individual														
<input checked="" type="checkbox"/> Asincrono	<input checked="" type="checkbox"/> Equipo de trabajo														
Evaluado por / Retroalimentado															
<input checked="" type="checkbox"/> Tutor	<input type="checkbox"/> Computador														
<input type="checkbox"/> Otros	<input checked="" type="checkbox"/> Foros														
	<input checked="" type="checkbox"/> Otros														
Calificación	La calificación dependerá de la ponderación que asigne el tutor en las preguntas														
Objetivo	Permite a los estudiantes aprender a dirigir y trabajar en grupo en base a un tema propuesto.														
Diagrama	<table border="0"> <thead> <tr> <th>Secuencia</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>1.- Actividad entregada</td> <td>Se verifica que el tutor haya levantado o puesto en sitio web (moodle) el documento o actividad de aprendizaje. Una vez verificado por el tutor, el estudiante descarga el documento para la revisión del mismo, y después se reúnen con el grupo de trabajo según sea el caso.</td> </tr> <tr> <td>2.- Mensaje o Explicación</td> <td>Se da un breve explicación sobre el temas al estudiante.</td> </tr> <tr> <td>3.- Asignación de tareas o preguntas</td> <td>Se asignas los lineamientos de la tarea que se encuentra documentado y entregado al estudiante con los temas.</td> </tr> <tr> <td>4.- Realización</td> <td>Los estudiantes realizan la actividad de aprendizaje de acuerdo a lo estipulado en el documento.</td> </tr> <tr> <td>5.- Entrega</td> <td>La entrega se lo realiza mediante la herramienta moodle, que le permitirá enviar el documento en formato Word.</td> </tr> <tr> <td>6.- Calificación</td> <td>Debido a que la herramienta moodle tienen muchas actividades, el tutor puede elegir diferentes maneras de calificar, mediante foros, taller, etc, que permitan verificar los conocimientos de cada uno de los estudiantes que conforman el grupo de trabajo</td> </tr> </tbody> </table>	Secuencia	Descripción	1.- Actividad entregada	Se verifica que el tutor haya levantado o puesto en sitio web (moodle) el documento o actividad de aprendizaje. Una vez verificado por el tutor, el estudiante descarga el documento para la revisión del mismo, y después se reúnen con el grupo de trabajo según sea el caso.	2.- Mensaje o Explicación	Se da un breve explicación sobre el temas al estudiante.	3.- Asignación de tareas o preguntas	Se asignas los lineamientos de la tarea que se encuentra documentado y entregado al estudiante con los temas.	4.- Realización	Los estudiantes realizan la actividad de aprendizaje de acuerdo a lo estipulado en el documento.	5.- Entrega	La entrega se lo realiza mediante la herramienta moodle, que le permitirá enviar el documento en formato Word.	6.- Calificación	Debido a que la herramienta moodle tienen muchas actividades, el tutor puede elegir diferentes maneras de calificar, mediante foros, taller, etc, que permitan verificar los conocimientos de cada uno de los estudiantes que conforman el grupo de trabajo
Secuencia	Descripción														
1.- Actividad entregada	Se verifica que el tutor haya levantado o puesto en sitio web (moodle) el documento o actividad de aprendizaje. Una vez verificado por el tutor, el estudiante descarga el documento para la revisión del mismo, y después se reúnen con el grupo de trabajo según sea el caso.														
2.- Mensaje o Explicación	Se da un breve explicación sobre el temas al estudiante.														
3.- Asignación de tareas o preguntas	Se asignas los lineamientos de la tarea que se encuentra documentado y entregado al estudiante con los temas.														
4.- Realización	Los estudiantes realizan la actividad de aprendizaje de acuerdo a lo estipulado en el documento.														
5.- Entrega	La entrega se lo realiza mediante la herramienta moodle, que le permitirá enviar el documento en formato Word.														
6.- Calificación	Debido a que la herramienta moodle tienen muchas actividades, el tutor puede elegir diferentes maneras de calificar, mediante foros, taller, etc, que permitan verificar los conocimientos de cada uno de los estudiantes que conforman el grupo de trabajo														

Tabla 3.46 Actividad de Aprendizaje

Diagrama de Actividad de Aprendizaje: Proyecto de Aplicación															
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software														
IMPLEMENTACIÓN	Departamento de Ciencias de Computación														
MATERIA	Ingeniería de Software I														
CREADORES	Alejandro Carrera, Gabriela del Castillo														
OBJETIVOS	Aplicar los conocimientos adquiridos en la lección actual mediante un ejercicio práctico														
SALIDAS	WORD, PDF														
DESCRIPCIÓN															
Tipo de Actividad	<table border="0"> <tr> <td>Entrenamiento</td> <td>Realizado por</td> </tr> <tr> <td><input type="checkbox"/> Sincrono</td> <td><input type="checkbox"/> Individual</td> </tr> <tr> <td><input checked="" type="checkbox"/> Asincrono</td> <td><input checked="" type="checkbox"/> Equipo de trabajo</td> </tr> <tr> <td>Evaluated por / Retroalimentado</td> <td></td> </tr> <tr> <td><input checked="" type="checkbox"/> Tutor</td> <td><input type="checkbox"/> Computador</td> </tr> <tr> <td><input type="checkbox"/> Otros</td> <td><input checked="" type="checkbox"/> Foros</td> </tr> <tr> <td></td> <td><input checked="" type="checkbox"/> Otros</td> </tr> </table>	Entrenamiento	Realizado por	<input type="checkbox"/> Sincrono	<input type="checkbox"/> Individual	<input checked="" type="checkbox"/> Asincrono	<input checked="" type="checkbox"/> Equipo de trabajo	Evaluated por / Retroalimentado		<input checked="" type="checkbox"/> Tutor	<input type="checkbox"/> Computador	<input type="checkbox"/> Otros	<input checked="" type="checkbox"/> Foros		<input checked="" type="checkbox"/> Otros
Entrenamiento	Realizado por														
<input type="checkbox"/> Sincrono	<input type="checkbox"/> Individual														
<input checked="" type="checkbox"/> Asincrono	<input checked="" type="checkbox"/> Equipo de trabajo														
Evaluated por / Retroalimentado															
<input checked="" type="checkbox"/> Tutor	<input type="checkbox"/> Computador														
<input type="checkbox"/> Otros	<input checked="" type="checkbox"/> Foros														
	<input checked="" type="checkbox"/> Otros														
Calificación	La calificación dependerá de la ponderación que asigne el tutor al proyecto de aplicación														
Objetivo	Permite a los estudiantes aprender a dirigir y trabajar en grupo en base a un proyecto de aplicación														
Diagrama	<table border="0"> <thead> <tr> <th>Secuencia</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>1.- Actividad entregada</td> <td>Se verifica que el tutor haya levantado o puesto en sitio web (moodle) el documento o proyecto de aplicación. Una vez verificado por el tutor, el estudiante descarga el documento para la revisión del mismo, y después se reúnen con el grupo de trabajo según sea el caso.</td> </tr> <tr> <td>2.- Mensaje o Explicación</td> <td>Se da un breve explicación sobre el temas al estudiante.</td> </tr> <tr> <td>3.- Asignación de tareas o preguntas</td> <td>Se asigna los lineamientos del proyecto de aplicación que se encuentra documentado y entregado al estudiante con el proyecto propuesto.</td> </tr> <tr> <td>4.- Realización</td> <td>Los estudiantes realizan el proyecto de aplicación de acuerdo a lo estipulado en el documento.</td> </tr> <tr> <td>5.- Entrega</td> <td>La entrega se lo realiza mediante la herramienta moodle, que le permitirá enviar el documento en formato Word.</td> </tr> <tr> <td>6.- Calificación</td> <td>Debido a que la herramienta moodle tienen muchas actividades, el tutor puede elegir diferentes maneras de calificar, mediante foros, taller, etc, que permitan verificar los conocimientos de cada uno de los estudiantes que conforman el grupo de trabajo</td> </tr> </tbody> </table>	Secuencia	Descripción	1.- Actividad entregada	Se verifica que el tutor haya levantado o puesto en sitio web (moodle) el documento o proyecto de aplicación. Una vez verificado por el tutor, el estudiante descarga el documento para la revisión del mismo, y después se reúnen con el grupo de trabajo según sea el caso.	2.- Mensaje o Explicación	Se da un breve explicación sobre el temas al estudiante.	3.- Asignación de tareas o preguntas	Se asigna los lineamientos del proyecto de aplicación que se encuentra documentado y entregado al estudiante con el proyecto propuesto.	4.- Realización	Los estudiantes realizan el proyecto de aplicación de acuerdo a lo estipulado en el documento.	5.- Entrega	La entrega se lo realiza mediante la herramienta moodle, que le permitirá enviar el documento en formato Word.	6.- Calificación	Debido a que la herramienta moodle tienen muchas actividades, el tutor puede elegir diferentes maneras de calificar, mediante foros, taller, etc, que permitan verificar los conocimientos de cada uno de los estudiantes que conforman el grupo de trabajo
Secuencia	Descripción														
1.- Actividad entregada	Se verifica que el tutor haya levantado o puesto en sitio web (moodle) el documento o proyecto de aplicación. Una vez verificado por el tutor, el estudiante descarga el documento para la revisión del mismo, y después se reúnen con el grupo de trabajo según sea el caso.														
2.- Mensaje o Explicación	Se da un breve explicación sobre el temas al estudiante.														
3.- Asignación de tareas o preguntas	Se asigna los lineamientos del proyecto de aplicación que se encuentra documentado y entregado al estudiante con el proyecto propuesto.														
4.- Realización	Los estudiantes realizan el proyecto de aplicación de acuerdo a lo estipulado en el documento.														
5.- Entrega	La entrega se lo realiza mediante la herramienta moodle, que le permitirá enviar el documento en formato Word.														
6.- Calificación	Debido a que la herramienta moodle tienen muchas actividades, el tutor puede elegir diferentes maneras de calificar, mediante foros, taller, etc, que permitan verificar los conocimientos de cada uno de los estudiantes que conforman el grupo de trabajo														

Tabla 3.47 Actividad de Aprendizaje

Diagrama de Actividad de Aprendizaje: Foros de Discusión													
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software												
IMPLEMENTACIÓN	Departamento de Ciencias de Computación												
MATERIA	Ingeniería de Software I												
CREADORES	Alejandro Carrera, Gabriela del Castillo												
OBJETIVOS	Valorar los criterios y conocimiento adquiridos en la lección mediante la discusión de un tema específico o propuesto de la lección.												
SALIDAS	Actividad Foros de Moodle.												
DESCRIPCION													
Tipo de Actividad	<p>Entrenamiento</p> <input type="checkbox"/> Sincrono <input checked="" type="checkbox"/> Asincrono												
	<p>Realizado por</p> <input checked="" type="checkbox"/> Individual <input type="checkbox"/> Equipo de trabajo												
	<p>Evaluated por / Retroalimentado</p> <input checked="" type="checkbox"/> Tutor <input checked="" type="checkbox"/> Otros												
	<input type="checkbox"/> Computador <input type="checkbox"/> Foros <input checked="" type="checkbox"/> Otros												
Calificación	La calificación dependerá de la ponderación que asigne el tutor al foro de discusión												
Objetivo	Permite a los estudiantes a ser capaz de discutir un tema en general mediante el análisis, comprensión, dominio del tema con un grupo de estudiantes.												
Diagrama	<table border="1"> <thead> <tr> <th>Secuencia</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>1.- Actividad entregada</td> <td>Se verifica que el tutor haya levantado o puesto en el sitio web (moodle) la actividad (Foro). Una vez verificado por el tutor, el estudiante verifica el tema a discutir.</td> </tr> <tr> <td>2.- Mensaje o Explicación</td> <td>Se da un breve explicación sobre como se realizará el foro de discusión y sus lineamientos.</td> </tr> <tr> <td>3.- Asignación de tareas o preguntas</td> <td>La tarea a realizar en el foro es la discusión del tema propuesto. Con algunas condiciones que el tutor proponga.</td> </tr> <tr> <td>4.- Realización</td> <td>Los estudiantes realizan el foro de acuerdo a el tiempo de duración que proponga el tutor, luego se cerrará el foro y los estudiantes no podrán participar en foro.</td> </tr> <tr> <td>5.- Calificación</td> <td>Debido a que la herramienta moodle (foro) nos permite asignar una calificación o no el tutor asignará la calificación adecuada según el criterio del estudiante.</td> </tr> </tbody> </table>	Secuencia	Descripción	1.- Actividad entregada	Se verifica que el tutor haya levantado o puesto en el sitio web (moodle) la actividad (Foro). Una vez verificado por el tutor, el estudiante verifica el tema a discutir.	2.- Mensaje o Explicación	Se da un breve explicación sobre como se realizará el foro de discusión y sus lineamientos.	3.- Asignación de tareas o preguntas	La tarea a realizar en el foro es la discusión del tema propuesto. Con algunas condiciones que el tutor proponga.	4.- Realización	Los estudiantes realizan el foro de acuerdo a el tiempo de duración que proponga el tutor, luego se cerrará el foro y los estudiantes no podrán participar en foro.	5.- Calificación	Debido a que la herramienta moodle (foro) nos permite asignar una calificación o no el tutor asignará la calificación adecuada según el criterio del estudiante.
Secuencia	Descripción												
1.- Actividad entregada	Se verifica que el tutor haya levantado o puesto en el sitio web (moodle) la actividad (Foro). Una vez verificado por el tutor, el estudiante verifica el tema a discutir.												
2.- Mensaje o Explicación	Se da un breve explicación sobre como se realizará el foro de discusión y sus lineamientos.												
3.- Asignación de tareas o preguntas	La tarea a realizar en el foro es la discusión del tema propuesto. Con algunas condiciones que el tutor proponga.												
4.- Realización	Los estudiantes realizan el foro de acuerdo a el tiempo de duración que proponga el tutor, luego se cerrará el foro y los estudiantes no podrán participar en foro.												
5.- Calificación	Debido a que la herramienta moodle (foro) nos permite asignar una calificación o no el tutor asignará la calificación adecuada según el criterio del estudiante.												

Tabla 3.26 Actividad de Aprendizaje

3.3.5. ESPECIFICACIÓN DEL FRAMEWORK

A pesar que el contenido es la parte más importante del curso, en nuestro caso no fue necesario la creación de un framework, por que este curso fue creado con propósito de utilizar una plataforma de educación virtual (Moodle).

A continuación mostramos conceptos básicos y descripción de un framework

3.3.5.1. Definición del frameWork

El Framework es todo aquello que no es contenido. Proporciona el ambiente para el contenido del curso. Contiene recursos y servicios, mecanismos de acceso tales como menú, índice, glosario, mapa del curso, motor de búsqueda.

Moodle incluye herramientas de colaboración como: Chat, foros, cuestionarios, talleres, consultas, diarios, glosarios, lecciones, etc. En Moodle incluye también herramientas administrativas para registro de usuarios, manejos de usuarios, manejo de calificaciones, monitoreo de progreso, etc.

3.3.5.2. Objetivo del FrameWork

El principal objetivo del frameWork es presentar el contenido al usuario, proporcionar un escenario amigable sobre el cual de presentará un curso. Ver figura 3.12

Moodle posee mecanismo de navegación, búsqueda, multimedia, y la capacidad de adaptar herramientas adicionales como radio, voz, video conferencia.

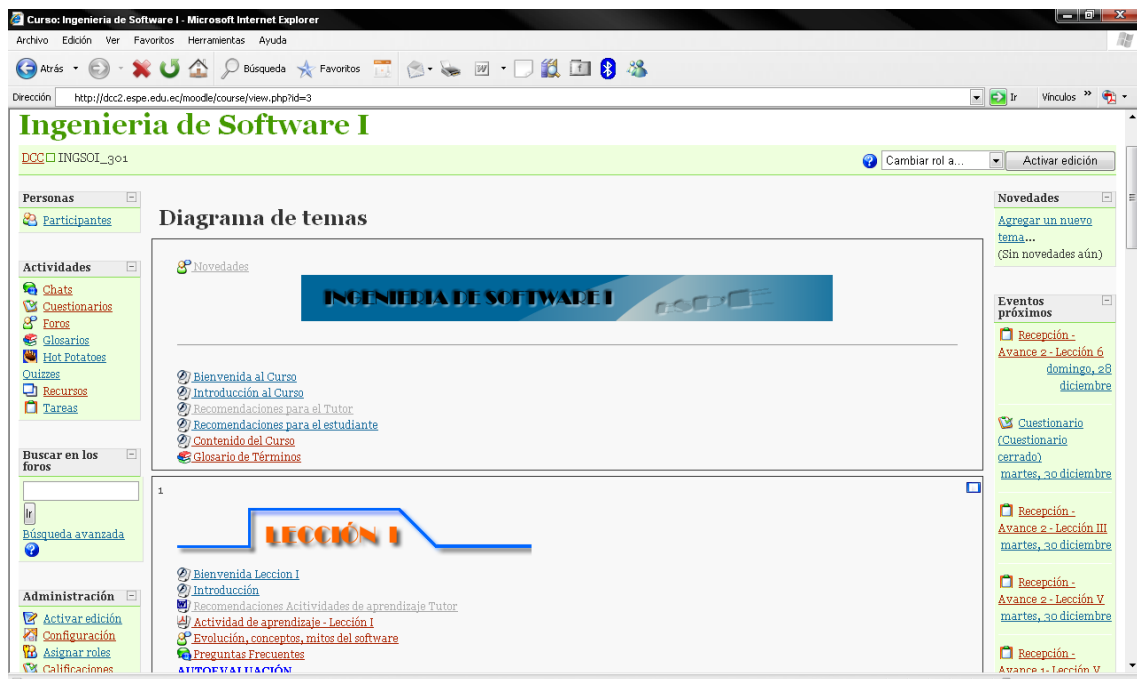


Figura 3.13 Framework Moodle

3.3.6. LISTADO DE LOS REQUERIMIENTOS TECNOLÓGICOS

Una vez que se haya definido el contenido y elegido el frameWork (Moodle) del curso se puede listar las herramientas y tecnología necesarias.

3.3.6.1. Participantes y procesos

Existen 3 grupos de personal involucrado en la realización del curso. El primer grupo corresponde a los productores del curso, el segundo grupo son los usuarios del curso y el tercer grupo corresponde al personal que va administrar y dar mantenimiento al servidor. En la elaboración de este curso existe otro grupo especial, que es el encargado de administrar la plataforma de educación Virtual, que por tratarse de una herramienta muy potente requiere de un administrador que tenga conocimientos y manejo de la plataforma (Moodle).

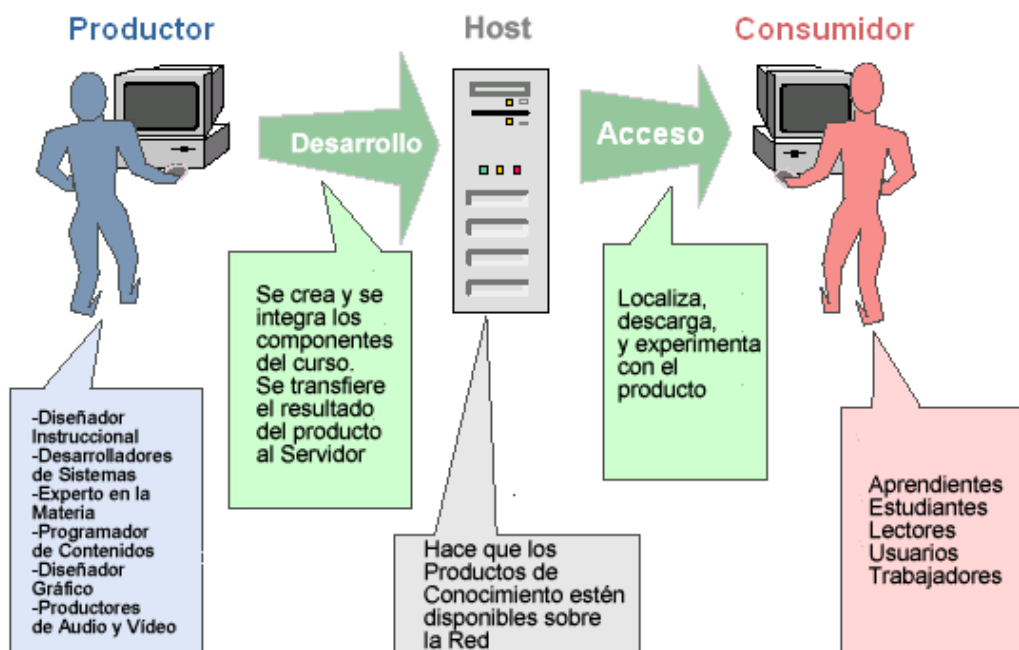


Figura 3.14. Participantes y Procesos. ^{3.7}

El proceso de creación de un curso es comúnmente conocido como Authoring. El proceso de consumir el curso es comúnmente conocido como Accesing.

Los productores incluyen a los diseñadores instruccionales, a los desarrolladores de sistemas, a los expertos den la metería, a los programadores de contenido, diseñadores gráficos, a los productores de audio y video y otras soluciones creativas.

Los usuarios del curso son conocidos como aprendientes, alumnos o estudiantes.

El administrador es el encargado de administrar y dar mantenimiento al curso. El hosting es el servidor donde radica o esta instalado la plataforma.

El proceso de authoring cubre las actividades de los productores, quienes crean e integran los componentes del curso y los transfieren al host.

Igualmente el Accesing, se refiere a las actividades que son realizadas por los usuarios quienes localizan, descargan y experimentan con el curso.

^{3.7} Grafico Tomado del Documento: "Developing Knowledge Products". William Horton Consulting.

3.3.6.2. Participantes y que necesitan ellos

Cada participante del proyecto (Productor, Host y Consumido) requerirá de un formulario de tecnología: Hardware, conexión de red y software (Ver figura3.15).

El host requerirá un servidor de red, se puede pensar en este como un poderoso servidor personal optimizado para entregar información en la red.

El host también requerirá de una conexión de red de alta velocidad para entregar la información simultáneamente a varios usuarios. Y el host requerirá de un software de Servidor Web, igualmente incluirá herramientas especiales de colaboración y servidores de media.

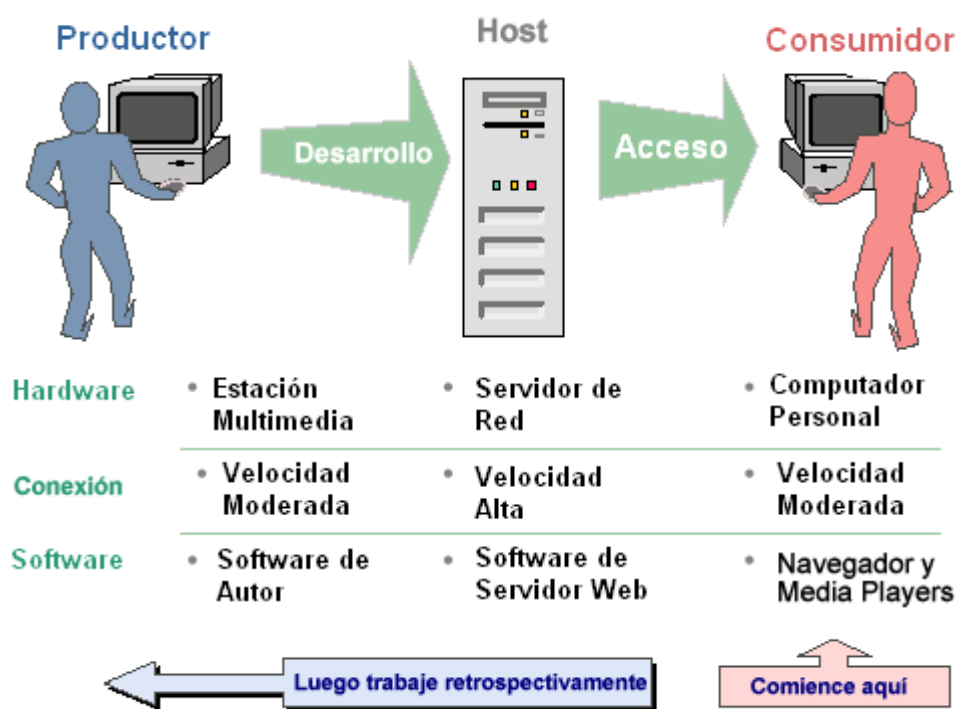


Figura 3.15. Necesidades de los Participantes.^{3.8}

El productor requerirá estaciones multimedia para realizar gráficas, animaciones, iconos, vídeo clips, sonidos y otra media necesaria. Típicamente una estación de trabajo costará dos veces más que el computador personal usado para el acceso al curso.

^{3.8} Tomado del Documento: "Developing Knowledge Products". William Horton Consulting.

Herramientas y tareas para cada Nivel

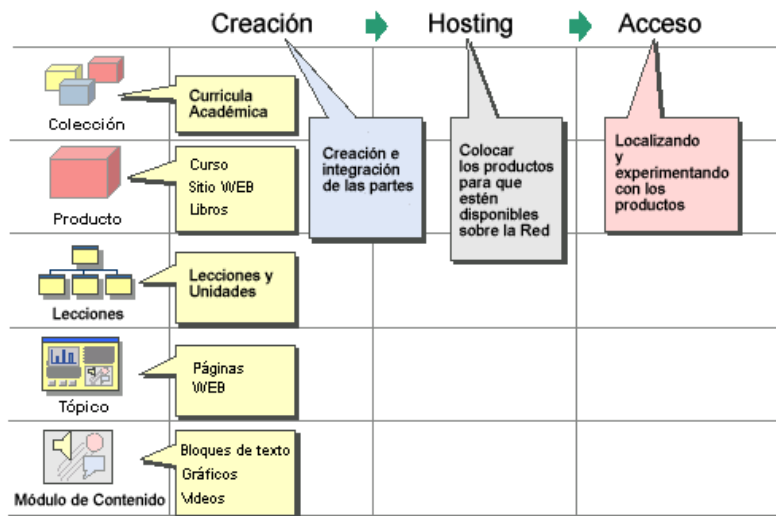


Figura 3.16 Herramientas y Tareas para Cada Nivel. ^{3.9}

Perpendicularmente están los niveles del curso o producto de conocimiento, en el nivel mas alto esta una colección el cual representa el conjunto de cursos e-learning. Debajo de este nivel de producto esta el objeto de conocimiento individual, conocido como curso en línea.

En el siguiente nivel se encuentra la unidad, esta unidad esta conformada por varias lecciones o capítulos, dentro de estos se encuentran, paginas, actividades, pruebas. Dentro de estos temas encontramos bloques de texto, animaciones, gráficos, imágenes, etc. Figura 3.16

Los procesos horizontales son autor, hosting y acceso del usuario. Autor involucra la creación e integración de componentes dentro del producto de conocimiento. El hosting es en donde la información se almacena y el acceso lo realiza el estudiante.

3.3.6.3. Herramientas para el curso

A continuación se va a detallar el software utilizado para la construcción del curso e-learning de la materia de "Ingeniería de Software I".

^{3.9} Tomado del Documento: "Developing Knowledge Products". William Horton Consulting.

- El browser o navegador que el usuario (alumno) va utilizar para entra en el sitio o Campus Virtual es Microsoft Internet Explorer.
- El hosting o servidor será el que utiliza la Escuela Politécnica del Ejercito
- Para la creación del contenido curso se utilizo las herramientas: *Macromedia Flash MX 2004* para las presentaciones del contenido de las lecciones, *Microsoft Word 2003* para describir los ejercicios de las lecciones, *Macromedia FireWorks* para crear y convertir las imágenes a archivos insertables de Flash.
- *The Interactive TTS Loquendo* para la realización de sonido de texto a voz.
- *GoldWave* para las secuencias de sonidos y conversión a formato a mp3.
- Moodle (Modular Object Oriented Distance Learning Enviroment) para el levantamiento del curso e-learning.

3.3.6.4. Especificación de formatos archivos

- Para la creación del contenido del curso y CD de apoyo se utilizo el Kit de Macromedia MX 2004

Tabla 3.48 formato de archivo

Tema	Programa	Extensión
Contenido del curso	Flash MX 2004	FLA, SWF
Actividad de Aprendizaje	Adobe Acrobat	.PDF
Resumen	Flash MX 2004	FLA, SWF
Reflexión	Flash MX 2004 o Microsoft Power Point 2003	FLA, SWF o .PPT
Evaluaciones	HotPotatoes 6	jcl , jcw, jmt, jmx, jqz, jc
Sonidos	Loquendo TTS	.wav
Conversor	GoldWave	.mp3

3.4. Etapa de construcción

Este proceso de construcción es una imagen del procesos de diseño, en este se sigue un esquema de trabajo desde arriba hacia abajo, desde conceptos extensos hasta llegar a minuciosos detalles. En al construcción se comienza con pequeños componente, primeros se construye los módulos de contenido, se los prueba y luego se les integra en

dentro de la paginas o presentaciones. Seguidamente se le pone a prueba individualmente y se le integra a cada una de las lecciones. Ver figura 3.17

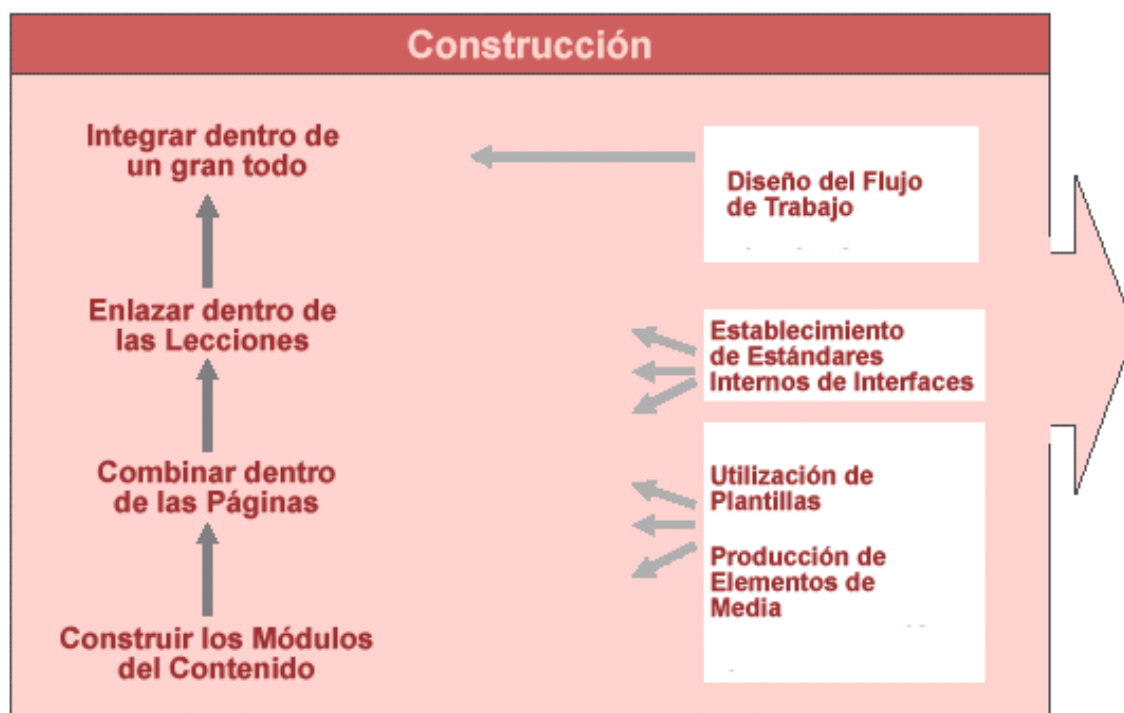


Figura 3.17 Construcción^{3.10}

En el desarrollo de los contenidos se debe tener en cuenta que el curso sea eficiente y consistente. La construcción de los módulos del contenido y la paginas deben estar estructuradas de manera que sean componente reutilizables tales como plantillas bases, secundarias.

En la elaboración de los contenidos se estableció estándares en lo referente a colores, tamaño, tipografía, botones, interactividad, etc.

La etapa de construcción consta de las siguientes fases:

1. Diseño del Flujo de Trabajo
2. Establecimiento de Estándares Internos
3. Utilización de Plantillas
4. Migración e Integración de Materiales

^{3.10} Grafico Tomado del Documento: "Developing Knowledge Products". William Horton Consulting.

3.4.1. DISEÑO DEL FLUJO DE TRABAJO

3.4.1.1. Flujo de Trabajo de Producción

Primero es necesario conocer los compromisos de los miembros del grupo de trabajo:

Los autores del curso:

- Experto en la materia
- El diseñador es quien es el responsable de la apariencia del las presentaciones del curso.
- Programador es quien escribe el código necesario para tareas técnicas complejas.
- Productor multimedia es quien hace las animaciones o objetos multimedia que contenga el curso.



Figura 3.18. Flujo de Trabajo de Producción.^{3.10}

^{3.10} Grafico Tomado del Documento: "Developing Knowledge Products". William Horton Consulting.

3.4.1.2.Utilización de plantillas

Se conoce como plantillas a los elementos de trabajo del producto que contiene espacios preestablecidos para introducir el contenido. Estos nos ayudan a dirigir y proporcionar formatos y toda la estructura que conforman la creación del contenido. Ver figura 3.19

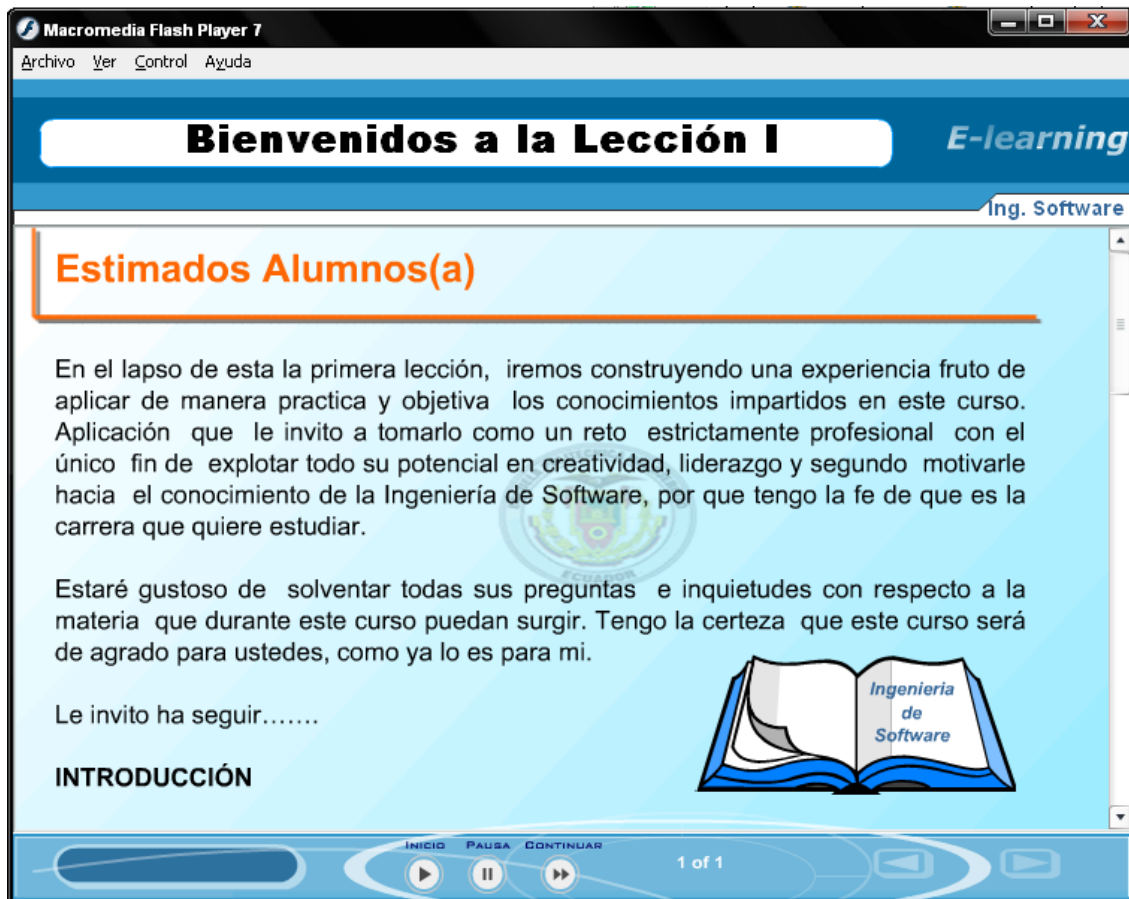
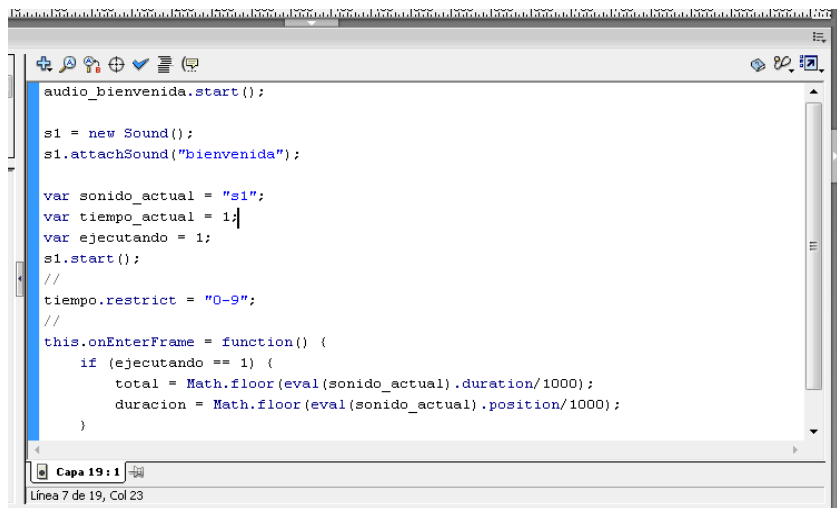


Figura 3.19. Plantilla de trabajo

3.4.1.3.Utilización de Scripts

Los scripts añaden interactividad, efectos, botones, sonidos al contenido del curso. Figura 3.20



```
audio_bienvenida.start();

s1 = new Sound();
s1.attachSound("bienvenida");

var sonido_actual = "s1";
var tiempo_actual = 1;
var ejecutando = 1;
s1.start();
//
tiempo.restrict = "0-9";
//
this.onEnterFrame = function() {
    if (ejecutando == 1) {
        total = Math.floor(eval(sonido_actual).duration/1000);
        duracion = Math.floor(eval(sonido_actual).position/1000);
    }
}
```

Capa 19: 1
Línea 7 de 19, Col 23

Figura 3.20. Script de programación ^{3.11}

Los script ayudan y dan la oportunidad de automatizar funciones mediante de programación, añadir sonido, control de botones. En nuestra figura anterior se muestra el código action script que es utilizado en macromedia flash y en ese ejemplo reutiliza para la inserción de un sonido a la presentación del objeto de conocimiento.

Conjuntamente se utilizó otro tipo de script, para el desarrollo de evaluaciones de las lecciones como es JavaScript que utiliza la herramienta Hotpotatoes para aislar o separa las respuestas de los test del curso.

3.4.2. ESTABLECIMIENTO DE ESTÁNDARES INTERNOS DE INTERFACES

Para conformar la uniformidad de las presentaciones creadas para las múltiples lecciones de los usuarios el grupo de trabajo utilizó diferentes herramientas y formatos de archivos.

Lo primero se estableció estándares internos para colores, tipografía, interactividad, sonido.

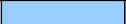

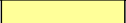



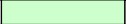

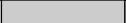
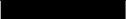
^{3.11} Script desarrollado por Alejandro Carrera

3.4.2.1. Selección de Colores

En general, se utilizaron los colores azul, celeste, sus tonalidades, blanco, negro y 6 colores adicionales.

Los colores se especifican en valores hexadecimales. De manera que los autores y diseñadores designan un color a cada de los diferentes elementos de la plantilla. Los colores que se utilizaron en el curso se especifican a continuación en el formulario CEL-IS-PCOLORES-01.

Tabla 3.49 Paleta de colores

PALETA DE COLORES						
NOMBRE DEL PROYECTO	Curso E-learning de la materia de Ingeniería de Software					
IMPLEMENTACIÓN	Departamento de Ciencias de Computación					
MATERIA	Ingeniería de Software I					
CREADORES	Alejandro Carrera, Gabriela del Castillo					
OBJETIVOS	Dar colores a los elementos, grafico del contenido del curso					
DESCRIPCION						
Nombre del color ó número	Ejemplo	Valores RGB			Valor Hexadec.	Donde es usado el color
		Rojo	Verde	Azul		
Azul1		103	204	255	99CCFF	Relleno de imagen, fondo de tabla
Azul2		102	153	204	6699CC	Relleno de paginas de Test
Azul3		51	102	153	336699	Cuadros de encabezados
Azul4		0	51	102	006699	Cuadros de encabezados
Amarillo1		255	255	204	FFFFCC	Fondos de Tablas
Amarillo2		255	255	153	FFFF99	No se utiliza
Amarillo3		255	255	102	FFFF66	Fondos de Tablas
Amarillo4		255	204	102	FF6600	Título de contenido
Rojo1		255	204	204	FFCCCC	No se utiliza
Rojo2		255	153	153	FF9999	No se utiliza
Rojo3		204	51	51	CC3333	Fondos de Tablas
Rojo4		153	51	51	993333	Fondos de Tablas
Verde1		204	255	204	CCFFCC	Fondo de documento
Verde2		153	204	153	99CC99	No se utiliza
Verde3		0	153	102	009966	No se utiliza
Verde4		0	102	51	006633	No se utiliza
Gray1		238	238	238	EEEEEE	Fondos de Tablas
Gray2		221	221	221	DDDDDD	No se utiliza
Gray3		204	204	204	CCCCCC	No se utiliza
Gray4		128	128	128	808080	Fondos de Tablas
Blanco		255	255	255	FFFFFF	Fondo de encabezado
Negro		0	0	0	000000	Texto, links, subindice

3.4.2.2. Selección del Tipo de Letra

Así mismo como los elementos o gráficos necesitan colores, los tipos de letra son utilizados y especificados de manera detallada en la tipografía del proyecto, de esta

manera el texto del objeto de conocimiento tiene sus especificación en la presentaciones de cada una de la lecciones, tales especificaciones conforma una plantilla básica para el curso formulario CEL-IS-TIPOGRAFÍA-01.

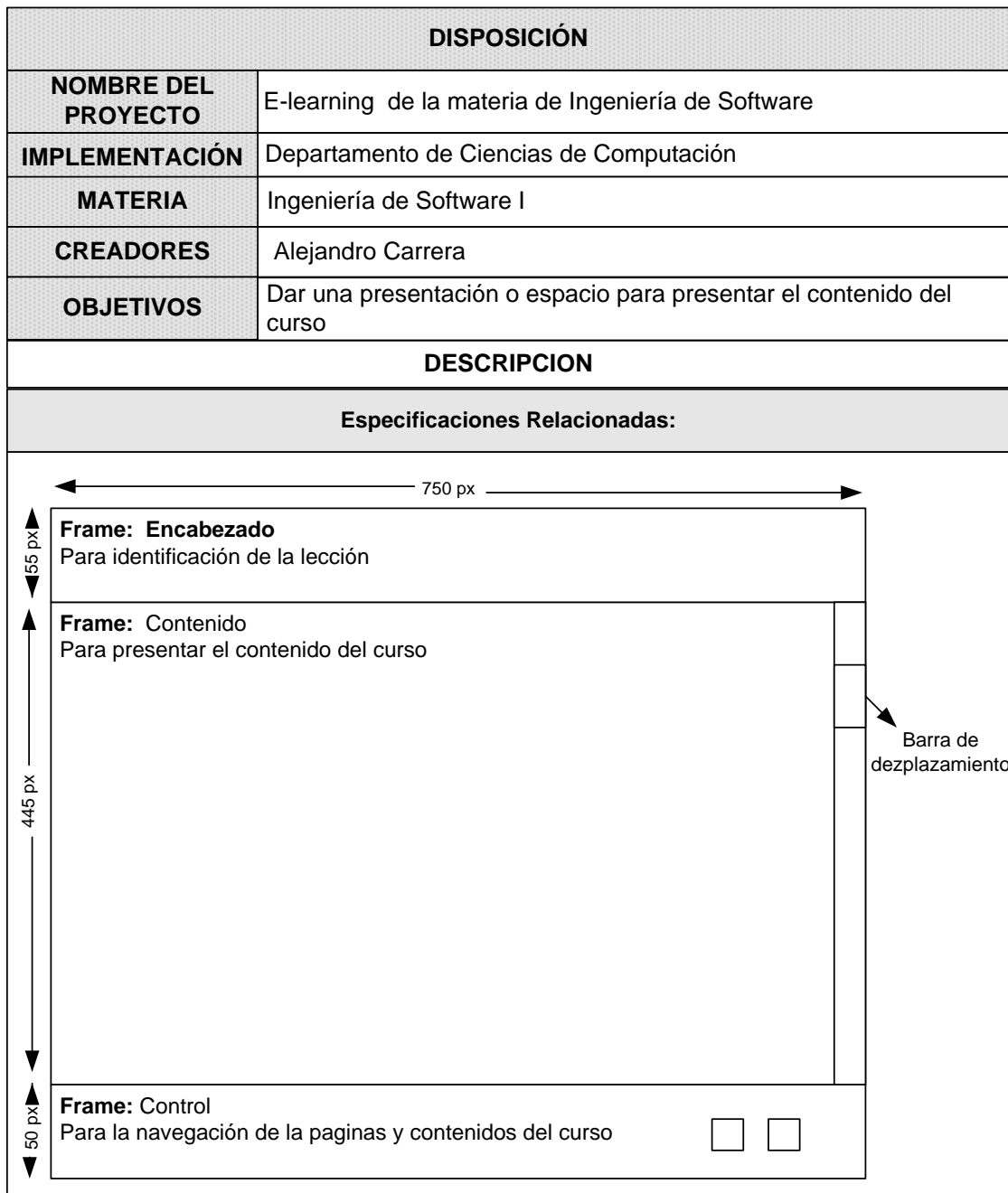
Tabla 3.50 Tipografía

TIPOGRAFIA						
NOMBRE DEL PROYECTO	E-learning de la materia de Ingeniería de Software					
IMPLEMENTACIÓN	Departamento de Ciencias de Computación					
MATERIA	Ingeniería de Software I					
CREADORES	Alejandro Carrera, Gabriela del Castillo					
OBJETIVOS	Dar un tipo de letra a los elementos, grafico del contenido del curso					
DESCRIPCION						
Componente	Especificaciones Relacionadas:					
	Fuente					
	Nombre	Tamaño	Estilo	Color	Fondo	Alineación
Título	ARIAL BLACK	26 ptos.	Negrita	#FF6600	(base)	Izquierda
Sub Título	ARIAL	27 ptos.	Negrita	#000000	(base)	Izquierda
Encabezado 1	ARIAL	17 ptos.	Negrita	(base)	(base)	Izquierda
Encabezado 2	ARIAL	17 ptos.	Negrita	(base)	(base)	Izquierda
Encabezado 3	ARIAL	17 ptos.	Negrita	(base)	(base)	Izquierda
Encabezado 4	ARIAL	17 ptos.	Negrita	(base)	(base)	Izquierda
Título de la Tabla	ARIAL	17 ptos.	(base)	(base)	(base)	Centrado
Cabecera Tabla	ARIAL	17 ptos.	(base)	#000000	(base)	Centrado
Fila Tabla	ARIAL	17 ptos.	(base)	#000000	(base)	Centrado
Texto	ARIAL	17 ptos.	(base)	#000000	(base)	Justificado

3.4.2.3. Disposición de Frame

El frame principal que se utiliza para las presentaciones se subdivide en: Cabecera, contenido y ubicación de botones de transición donde en cada una se especifica el lugar. Para nuestro proyecto se ha elaborado un frame para las presentaciones y contenido del curso, aprovechando las características que ofrece Moodle y seguidamente para el CD de apoyo de la materia. Formulario CEL-IS-DISPOSICIÓN-01.

Tabla 3.51 Frame



3.4.3. UTILIZACIÓN DE PLANTILLAS

Las plantillas son modelos de páginas de trabajo para creación de las páginas de los contenidos de las lecciones, recordemos que esta plantilla será como base para todas las lecciones y así simplificamos la producción masiva. A continuación se presenta la plantilla general para las lecciones. Figura 3.21



Figura 3.21 Plantilla para una página

La plantilla general reduce enormemente la cantidad de páginas, debido a que usa la plantilla modelo básico del curso. A continuación se muestra un ejemplo de una plantilla con la forma básica. Figura 3.22

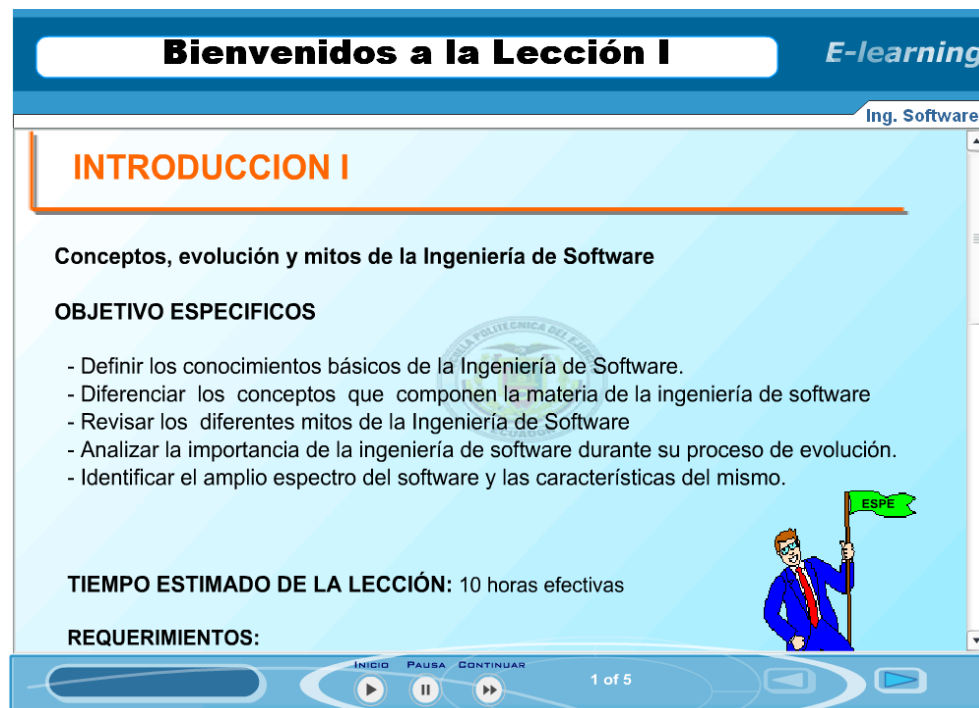


Figura 3.22 Uso de la Plantilla General

3.4.3.1. Proceso para la creación de una plantilla

Para utilizar efectivamente las plantillas hay que realizar cuidadosamente el diseño de las mismas. En lugar de crear una página única desde el principio, se debe seguir un esquema en el cual se cree páginas únicas mediante la combinación de plantillas.

El proyecto comienza con una plantilla base. La plantilla base especifica las cosas que son comunes para todas las páginas. Esta plantilla puede incluir un fondo y un contenido específico que va a aparecer en todas las páginas. Ver Figura 3.23

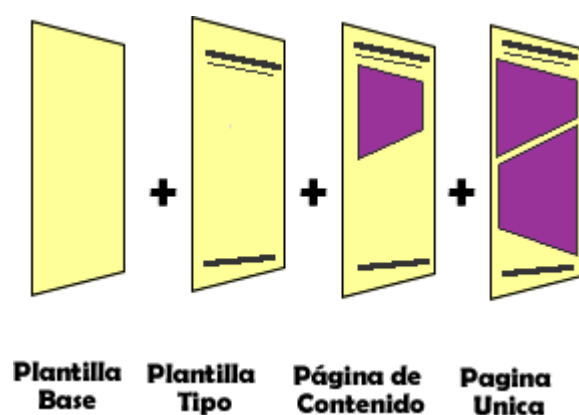


Figura 3.23 Base para las plantillas^{3.12}

Para páginas comunes se puede utilizar la plantilla base, adicionalmente se puede aumentar o quitar algún elemento no común y que solo afecte en un porcentaje bajo. La página resultante es un producto de la combinación de la plantilla base y pantalla tipo es una plantilla secundaria.

3.4.3.2. Plantillas necesarias

Las páginas que necesitan plantillas para la construcción del presente proyecto son:

- Página de Bienvenida
- Página de Introducción
- Página de Resumen

^{3.12} Adaptado del Documento: "Developing Knowledge Products". William Horton Consulting.

- Pagina de Sitios Recomendados
- Pagina de Reflexión

3.4.3.3.Plantillas para las páginas de Evaluación

Para las evaluaciones las páginas están conformadas por dos grupos que son: las plantillas de las actividades de aprendizaje y las plantillas de los test, a continuación se presenta una plantilla de cada una de los grupos. Cabe destacar que se elaboró una plantilla única para todas las lecciones.

Tabla 3.52. Página Bienvenida

Especificación de la Plantilla	
Tipo de Tópico	Bienvenida a la Lección N°
Plantilla Base	Plantilla Tipo
<ul style="list-style-type: none"> ◆ Banner del Curso ◆ Botones de Navegación de la Unidad ◆ Nombre de la materia ◆ Numero de lección ◆ Caja de numero de paginas ◆ Botón inicio(sonido) ◆ Botón pausa(sonido) ◆ Botón continuar(sonido) ◆ Caja de información 	<ul style="list-style-type: none"> ▪ Mensaje al estudiante ▪ Introducción ▪ Objetivo de la lección ▪ Contenidos ▪ Actividades Tareas ▪ Recursos ▪ Evaluación y retroalimentación
Página Terminada	

Bienvenidos a la Lección I
E-learning

Ing. Software


Estimados Alumnos(a)

En el lapso de esta la primera lección, iremos construyendo una experiencia fruto de aplicar de manera practica y objetiva los conocimientos impartidos en este curso. Aplicación que le invito a tomarlo como un reto estrictamente profesional con el único fin de explotar todo su potencial en creatividad, liderazgo y segundo motivarle hacia el conocimiento de la Ingeniería de Software, por que tengo la fe de que es la carrera que quiere estudiar.

Estaré gustoso de solventar todas sus preguntas e inquietudes con respecto a la materia que durante este curso puedan surgir. Tengo la certeza que este curso será de agrado para ustedes, como ya lo es para mi.

Le invito ha seguir.....

INTRODUCCIÓN



INICIO PAUSA CONTINUAR
1 of 1

Tabla 3.53. Página Bienvenida

Especificación de la Plantilla	
Tipo de Tópico	Introducción a la Lección N°
Plantilla Base	Plantilla Tipo
<ul style="list-style-type: none"> ◆ Banner del Curso ◆ Botones de Navegación de la Unidad ◆ Nombre de la materia ◆ Numero de lección ◆ Caja de numero de paginas ◆ Botón inicio(sonido) ◆ Botón pausa(sonido) ◆ Botón continuar(sonido) ◆ Caja de información 	<ul style="list-style-type: none"> ▪ Título de la lección ▪ Objetivos específicos ▪ Tiempo estimado de la lección ▪ Requerimientos de la lección ▪ Grafico y mensaje de la lección ▪ Materia de la lección
Página Terminada	

Bienvenidos a la Lección I
E-learning

Ing. Software

INTRODUCCION I


Conceptos, evolución y mitos de la Ingeniería de Software

OBJETIVO ESPECIFICOS

- Definir los conocimientos básicos de la Ingeniería de Software.
- Diferenciar los conceptos que componen la materia de la ingeniería de software
- Revisar los diferentes mitos de la Ingeniería de Software
- Analizar la importancia de la ingeniería de software durante su proceso de evolución.
- Identificar el amplio espectro del software y las características del mismo.

TIEMPO ESTIMADO DE LA LECCIÓN: 10 horas efectivas

REQUERIMIENTOS:



INICIO PAUSA CONTINUAR
1 of 5

Tabla 3.54. Página de Resumen

Especificación de la Plantilla	
Tipo de Tópico	Resumen de la Lección N°
Plantilla Base	Plantilla Tipo
<ul style="list-style-type: none"> ◆ Banner del Curso ◆ Botones de Navegación de la Unidad ◆ Nombre de la materia ◆ Numero de lección ◆ Caja de numero de paginas ◆ Botón inicio(sonido) ◆ Botón pausa(sonido) ◆ Botón continuar(sonido) ◆ Caja de información 	<ul style="list-style-type: none"> ▪ Título Resumen Lección ▪ Gráfico ▪ Tema de resumen
<i>Página Terminada</i>	


Bienvenidos a la Lección I
learning

Ing. Software

RESUMEN DE LA LECCIÓN I

Introducción a la Ingeniería de Software I

“Es establecer y usar principios de ingeniería orientada a obtener un software de manera económica, fiable, que funcione eficazmente sobre máquinas reales” (Fritz Bauer).



La Ingeniería de software a través de su evolución ha creado la necesidad de aplicar teorías, métodos y herramientas para el desarrollo, administración y mantenimiento de productos de software, el cambio de tecnología hace que aparezca en el mercado nuevas herramientas, lo cual hace que se realicen cambios rigurosos en pos de mejorar y reducir los tiempos de entrega.

El software no solo consiste en programas, documentación y líneas de código, también se considera atributos de los productos como son: mantenimiento, dependencia, eficiencia y uso, contribuyendo a la calidad del producto software.

INICIO PAUSA CONTINUAR
1 of 2

Tabla 3.55. Página de Reflexión

Especificación de la Plantilla	
Tipo de Tópico	Reflexión de la Lección N°
Plantilla Base	Plantilla Tipo
<ul style="list-style-type: none"> ◆ Banner del Curso ◆ Botones de Navegación de la Unidad ◆ Nombre de la materia ◆ Numero de lección ◆ Caja de numero de paginas ◆ Caja de información 	<ul style="list-style-type: none"> ▪ Titulo Reflexión Lección ▪ Gráfico ▪ Texto reflexión
Página Terminada	

REFLEXION
Learning

Ing. Software

Amor Eterno

El día que mi hija Maria José nació, realmente no sentí gran alegría por que la decepción que tenía era mayor que el gran acontecimiento que representa tener el primer hijo. Yo ansiaba tener un varón. A los dos días de haber nacido, fui a buscar a mis dos mujeres (mi esposa y mi hija), una lucia pálida y la otra radiante y dormilona.


1 of 9

Tabla 3.56. Página de Actividad de Aprendizaje

Especificación de la Plantilla	
Tipo de Tópico	Actividades de aprendizaje de la Lección N°
Plantilla Base	Plantilla Tipo
<ul style="list-style-type: none"> ◆ Carátula del ESPE. ◆ Mensaje al estudiantes. ◆ Titulo del número de la actividad didáctica. ◆ Objetivos Generales. ◆ Objetivos específicos. ◆ Accesorio didáctica. ◆ Instrucciones y grupos de trabajo. ◆ Ponderación y calificaciones. ◆ Lecturas requeridas 	<ul style="list-style-type: none"> ▪ Tema o tareas ▪ Ejercicios
Página Terminada	

BIENVENIDO SEÑOR ESTUDIANTE:

Al iniciar los estudios de esta asignatura, usted ya conoce las características del estudio vía E-learning de esta Universidad. Se le solicita de la manera más encarecida su responsable dedicación al estudio de la materia. Su disciplina en las lecciones es importante. Intentamos, en esta materia, ayudarle día a día, en la aprehensión y apropiación del conocimiento. Queremos que usted desarrolle habilidades y destrezas propias de las metodologías de un Ingeniero de Sistemas.

Actividades de Aprendizaje N° 1 

1. TAREA TEMA: ANÁLISIS Y EVOLUCIÓN DEL SOFTWARE, MITOS DEL SOFTWARE E IMPORTANCIA COMO PARTE DE UN PROCESO DE EVOLUCIÓN DE LA INGENIERÍA DE SOFTWARE.

1.1. Muchos problemas han ocurrido durante la era de la información en la cual se derivaron problemas, y una de ellas es la crisis del software que se caracterizó por no cumplir con los requerimientos del cliente, fallas frecuentemente, costos, etc.

1.1.1. Mencione estos problemas o características en cada una las épocas en que evolucionó el software desde 1950 hasta 2000 y de varios ejemplos positivos y negativos que indique el impacto que tuvo en nuestra sociedad (al menos 6 ejemplos).

1.2. Los mitos del software propagaron información errónea y confusión, lo que conllevó a otro de los problemas de la crisis del software durante los primeros años y hasta los actuales momentos. A continuación analice y describa la *realidad* para cada uno de los mitos descriptos a continuación.

1.2.1. Se decidió subcontratar el proyecto de software a un tercero, se puede relajar y dejar que esa compañía lo construya.

1.2.2. Se posee un libro lleno de estándares y procedimientos para construir un software. ¿Con esto basta para que mi personal construya el sistema?

1.2.3. Mi personal dispone de las herramientas de desarrollo de software más avanzadas en la actualidad, se les proporcionó de computadores de última generación. Estos influyen en el desarrollo?

1.2.4. Dentro de los requerimientos de un cliente, nos encontramos con la necesidad de cambiar una parte del sistema: Es fácil cambiar?

1.2.5. Desde la automatización de los procesos, el computador reemplazó en muchas compañías los procesos manuales elaborados por el hombre por sistemas automatizados. Estos proveen más confiabilidad que los procesos manuales?

1.2.6. Mayor confiabilidad del software aumenta la seguridad.

Tabla 3.57. Página de Actividad de Aprendizaje

Especificación de la Plantilla: Opción múltiple/Verdadero o falso	
Tipo de Tópico	Evaluaciones de la Lección N°
Plantilla Base	Plantilla Tipo
<ul style="list-style-type: none"> ◆ Encabezado ESPE. ◆ Departamento de ciencias de la Computación. ◆ Nombre de la unidad o lección ◆ Tiempo ◆ pregunta ◆ Botón siguiente pregunta ◆ Botón mostrar todas la preguntas. ◆ Botón index ◆ Botón siguiente ejercicio ◆ Lecturas requeridas 	<ul style="list-style-type: none"> ■ Preguntas ■ Ejercicios

Página Terminada

ESCUELA POLITÉCNICA DEL EJÉRCITO
Departamento de Ciencias de la Computación
Fundamentos de la Ingeniería de Software - Evaluación

9:41

Módulo sobre la realización de la evaluación

• Cada pregunta tiene reflejado un porcentaje e sobre el total del examen.

• Este examen supone a su vez el 60% de la nota final de la asignatura.

En las cuestiones en las que hay que marcar verdadero/falso, escoger un repuesto, seleccionar una serie de opciones, cada opción incorrecta tiene una penalización de del 80% de valor total de la pregunta es de así el valor de penalización de 1 su factor de penalización es 0.8

Show all questions

1 / 11 [Siguiente Pregunta](#)

¿Cual de los siguientes características es correcta a la crisis del software.?

A. • Imprecisión en la planificación del proyecto y estimación de los costos.

B. • Facilitar el control del proceso de desarrollo de software.

C. • No existen piezas de repuesto.

[Index](#) =>

Tabla 3.58. Página de Actividad de Aprendizaje

Especificación de la Plantilla: Crucigrama	
Tipo de Tópico	Evaluaciones de la Lección N°
Plantilla Base	Plantilla Tipo
<ul style="list-style-type: none">◆ Encabezado ESPE.◆ Departamento de ciencias de la Computación.◆ Nombre de la unidad o lección◆ Tiempo◆ pregunta◆ Botón index	<ul style="list-style-type: none">■ Ejercicios■ Preguntas horizontales■ Preguntas Verticales
Página Terminada	

ESCUELA POLITECNICA DEL EJERCITO
 Departamento de Ciencias de la Computación
 Fundamentos de la Ingeniería de Software - Evaluación

9:28

Nota sobre la penalización de la evaluación

- Cada pregunta tiene reflejado un porcentaje e sobre el total del examen.
- Este examen supone a su vez el 60% de la nota final de la asignatura.

En las cuestiones en las que hay que marcar verdadero/falso, escoger un repuesta, seleccionar una serie de opciones, cada opción incorrecta tiene una penalización de del 80% de valor total de la pregunta es de así el valor de penalización de 1 su factor de penalización es 0.8

Complete el crucigrama, pulse "Comprobar" para comprobar su respuesta. Si no esta seguro haga click en "Pista" paa obtener una ayuda. Pero recuerde que esto le bajara el puntaje de su calificación.

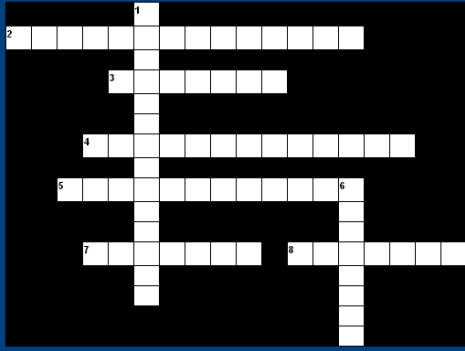


Tabla 3.59. Página de Actividad de Aprendizaje

Especificación de la Plantilla: Cuestionario Tipo Moodle	
Tipo de Tópico	Evaluaciones de la Lección N°
Plantilla Base	Plantilla Tipo
<ul style="list-style-type: none"> ◆ Título ◆ Tiempo 	<ul style="list-style-type: none"> ▪ Tema o tareas ▪ Ejercicios
<i>Página Terminada</i>	

Evaluación - Tipo Moodle - Intento 1

1 **El objetivo de PROCESO DE DESARROLLO DE DOCUMENTACION es:**
Puntos: -/1 Seleccione una respuesta.

Tiempo restante
0:04:20

a. Es el conjunto de actividades que planifican, diseñan, implementan, editan, producen, distribuyen y mantienen la información necesaria para los desarrolladores y los usuarios.

b. Abarca la planificación y la realización de todas las tareas de verificación.

c. Involucra la operación del sistema por parte del usuario y el soporte continuo.

2 **Herramientas.** -definen la secuencia en la que se aplican los métodos, las entregas, los controles de calidad y guías para evaluación del progreso
Puntos: -/1 Seleccione una respuesta.

a. Falso

b. Verdadero

3 **¿Cual de los siguientes características es correcta a la crisis del software.?**
Puntos: -/1 Seleccione una respuesta.

a. Facilitar el control del proceso de desarrollo de software.

b. No existen piezas de repuesto.

c. Imprecisión en la planificación del proyecto y estimación de los costos.

4 **Existen dos tipos de desarrollo evolutivo:** Desarrollo Exploratorio y Enfoque utilizando prototipos.
Puntos: -/1 Seleccione una respuesta.

a. Falso

b. Verdadero

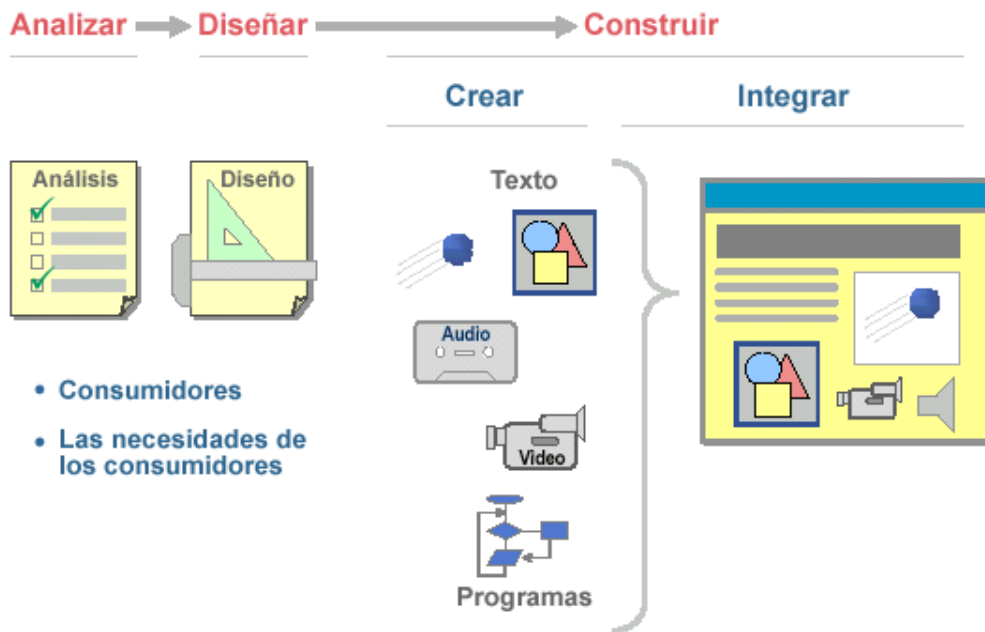
5 **Confiable** el software no debe causar daños físicos o económicos en el caso de fallos.
Es una características de:
A
Puntos: -/1 Seleccione una respuesta.

a. Características del software

0:04:20 Intranet local

3.4.3.4. Ensamblaje del Curso

Una vez que todos los materiales están en formato electrónico y los elementos de media han sido creados, se procedió a integrarlos en el curso dependiendo de la estructuración de las lecciones realizadas en la etapa de diseño. Ver Figura 3.24.



F

figura 3.24. Ensamblaje del Curso.^{3.13}

3.5. ETAPA DE EVALUACIÓN

La evaluación es el proceso en donde se valora un curso ó producto de conocimiento. Esta etapa es muy importante porque el grupo de trabajo se informa si los esfuerzos realizados valieron o no la pena y además cómo se puede mejorar y reforzar el curso ó curso ó producto de conocimiento.

3.5.1. Visión general del proceso de evaluación

Esta etapa se aplica principalmente al final del desarrollo de un curso, en este caso quedará planteado con el fin de que se realice un proceso de evaluación como otro proyecto de tesis. A continuación se detallarán los pasos de este proceso.

Para evaluar un curso ó producto de conocimiento se sigue un flujo sistemático de actividades: Ver Figura 3.25.

^{3.13} Gráfico Tomado del Documento: “Developing Knowledge Products”. William Horton Consulting.

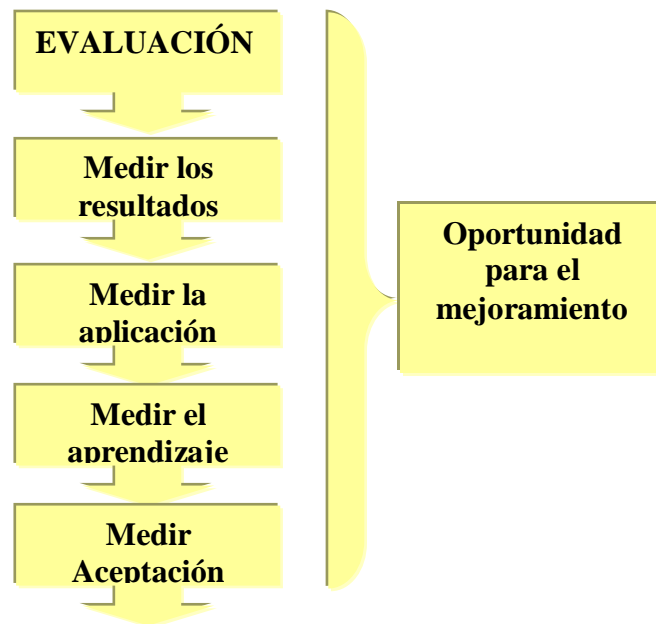


Figura 3.25. Proceso de Evaluación. ^{3.14}

El primer paso consiste en desarrollar un plan de evaluación. El próximo paso deberá ser el establecimiento de una línea base de medición del rendimiento de los usuarios, para lo cual se va a medir el nivel actual de rendimiento y ejecución del trabajo.

Aquí se establecerá una norma contra la cual se mida las mejoras. Igualmente, se podrá medir el nivel actual de conocimiento de los usuarios. En el plan de tesis se planteo que la tesis terminara cuando el curso este levantado en la plataforma de educación Virtual del departamento de Ciencias de Computación por tanto la evaluación terminara con la revisión y comprobación por parte del director de la tesis el Ing. Edgar Hermosa.

^{3.14} Gráfico adaptado del Documento: “Developing Knowledge Products”. William Horton Consulting.

CAPÍTULO IV

4.4. Levantamiento del curso en la Plataforma e-learning

Para el levantamiento del curso de Ingeniería de software se utilizó una computadora y la plataforma de educación Virtual Moodle.

Para esta demostración se realizó desde la creación del curso hasta el levantamiento de una lección.

4.4.1. Creación del curso

El primer paso es la creación del curso de Ingeniería de software no dirigimos a la plataforma e ingresamos con una cuenta de administrador, para la demostración utilizamos la cuenta de el Ing. Edgar hermosa proveída para la creación del curso.

The image shows a Moodle login interface with two main sections:

- Usuarios registrados:** A form for logging in with fields for 'Nombre de usuario' (containing 'ehermosa') and 'Contraseña' (masked with dots), and an 'Entrar' button. Below this, there is a link for 'Entrar como invitado' and a link for 'Sí, ayúdeme a entrar'.
- Registrarse como usuario:** A section with a welcome message and a 7-step registration process. At the bottom, there is a button labeled 'Comience ahora creando una cuenta'.

Figura 4.1 Login de usuario



Nos dirigimos a Administración del Sitio luego a Cursos y elegimos a Agregar/editar cursos

Después de esta imagen, enseguida aparece una pantalla donde ingresaremos los datos o configuración del curso. En esta pantalla encontraremos algunas propiedades muy importantes que se debe elegir de manera correcta. A continuación detallamos cada uno de estas opciones

Figura 4.2 Agregar cursos

Editar la configuración del curso

Ajustes generales

Categoría ? SOFTWARE

Nombre completo ? Nombre del curso

Nombre corto ? ID del curso

Número ID del curso ?

Resumen ?

Trebuchet 1 (8 pt) Idioma **B** *I* U ~~S~~ x₂ x₃

Escriba aquí la descripción del curso

Ruta:

Figura 4.3 Datos del cursos

Categoría.- aquí se elige la categoría a la que pertenece nuestro curso.

Nombre completo.- aquí se escribe el nombre completo del curso.

Nombre corto.- Muchas instituciones asignan nombres cortos a sus cursos para efectos administrativos, como BP102 o 4H-LIT. Para nuestro curso será INGSOI_301.

Número ID del Curso.- El número ID de un curso únicamente se usa cuando se compara este curso contra un sistema externo - lo cual nunca es mostrado dentro de Moodle.

Resumen.- Aquí se escribe una breve descripción del curso que se va a presentar.

Formato ? Formato semanal

Número de semanas o temas 10

Fecha de inicio del curso ? 11 diciembre 2008

Temas ocultos ? Las secciones ocultas se muestran en forma colapsada

Items de noticias para ver ? 5

Mostrar calificaciones ? Sí

Mostrar informes de actividad ? No

Tamaño máximo para archivos cargados por usuarios ? 16Mb

¿Es éste un metacurso? ? No

Figura 4.4 Datos del cursos

Formato: brinda la posibilidad de seleccionar entre tres opciones posibles: en nuestro caso es:

Número de semanas/temas: este parámetro sólo se aplica a los formatos “semanal” y por “temas”. En el formato por temas, se especifica el número de temas del curso.

Formato por temas: es muy parecido al formato semanal, excepto que en lugar de estar organizado por semana lo está por tema. Éstos no se encuentran limitados por el tiempo, por lo que no hay que especificar fechas de comienzo y fin. El cronograma es manejado por el tutor de acuerdo a su planificación personal.

Fecha de inicio del curso: aquí es donde se coloca la fecha de inicio de clases. Si usted está utilizando el formato semanal, esto afectará la organización de las semanas.

Mostrar calificaciones: esta opción le permite mostrar o no a los alumnos las notas de sus evaluaciones.

Informes de Actividades.- Los informes de actividades están disponibles para todos los participantes que muestren sus actividades en el presente curso. Al igual que las listas de contribuciones, estos informes incluyen un detallado registro de los accesos.

Tamaño máximo de los archivos.- Esta variable determina el tamaño máximo permitido para los archivos subidos por los estudiantes a este curso, limitado por el tamaño máximo del sitio, que es determinado por el administrador.

Rol por defecto.- Esta opción determina si los estudiantes son capaces de matricularse a sí mismos usando el conector interactivo de matrícula por defecto (por ejemplo, matrícula interna) para el curso, o si ellos obtendrían el mensaje: "Este curso no es matriculable por el momento".

Disponibilidad del curso.- Esta opción permite "esconder" su curso.

Clave de acceso al curso.- Una contraseña de acceso sirve para restringir el acceso a personas ajenas a ese curso.

Luego de haber terminado de colocar la configuración del curso, inmediatamente aparecerá la paginas principal del curso

4.5. Levantamiento del inicio del curso

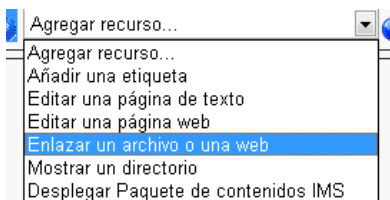
El inicio del curso esta conformado por las siguientes presentaciones:

- Bienvenida al Curso archivo
- Introducción al Curso archivo
- Recomendaciones para el Tutor archivo
- Recomendaciones para el estudiante archivo
- Contenido del Curso archivo
- Glosario de Términos

A continuación mostraremos como se levanta una presentación.

Lo primero que debemos hacer es activar la edición hace clic en el botón

Activar edición



Luego escogemos agregar recurso y la opción *enlazar un archivo a una Web* y continuación saldrá una cuadro de dialogo que presentamos a continuación

Figura 4.5 Enlazar una archivo a una Web.

4.6. Levantamiento del de un documento

Una vez cuando hagamos clic en Enlazar un archivo o una Web tendremos la siguiente pantalla que mostramos a continuación

Ajustes generales

Nombre *

Resumen ?

Trebuchet 1 (8 pt) Idioma **B** *I* U ~~S~~

Ruta:

Enlazar un archivo o una web

Ubicación

Figura 4.6 Datos para enlazar un archivo

Nombre.- en esta opción se inserta el nombre que va a llevar en la presentación del recurso.

Resumen.- en esta opción se inserta indicaciones que tutor desee informar a los estudiantes.

Ubicación.- en esta opción se inserta la ubicación exacta donde se encuentra el archivo que se quiere levantar en el sitio. Cabe acotar que al elegir subir un archivo se debe seguir unos pasos e indicaciones siguientes.

INGSOI » Archivos

	Nombre	Tamaño	Modificado	Acción
<input type="checkbox"/>	ing_inicio	9.7Mb	4 dic 2008, 09:06	Renombrar
<input type="checkbox"/>	ing_leccion_L1	10Mb	4 dic 2008, 11:28	Renombrar
<input type="checkbox"/>	ing_leccion_L2	14Mb	4 dic 2008, 10:29	Renombrar
<input type="checkbox"/>	ing_leccion_L3	4.2Mb	4 dic 2008, 11:08	Renombrar
<input type="checkbox"/>	ing_leccion_L4	7.5Mb	4 dic 2008, 11:38	Renombrar
<input type="checkbox"/>	ing_leccion_L5	10.6Mb	4 dic 2008, 02:50	Renombrar
<input type="checkbox"/>	ing_leccion_L6	4.6Mb	4 dic 2008, 04:22	Renombrar
<input type="checkbox"/>	ing_leccion_L7	2.1Mb	4 dic 2008, 06:11	Renombrar
<input type="checkbox"/>	prueba	20.7Mb	3 dic 2008, 02:10	Renombrar
<input type="checkbox"/>	FONDO.png	59.4Kb	11 nov 2008, 06:31	Elegir Renombrar
<input type="checkbox"/>	ingLogo.png	145.6Kb	26 nov 2008, 04:40	Elegir Renombrar

Con los archivos escogidos... ▾

Figura 4.7 Datos para enlazar un archivo

- Antes de subir un archivo en el sitio se recomienda crear una carpeta especificando el contenido que va introducir, por la razón de llevar ordenadamente el material del curso.
- En nuestro ejemplo elegimos la carpeta [ing inicio en esta carpeta se encuentra el archivo que vamos levantar en nuestro sitio.](#)
- [Cuando elegimos el archivo \(bienvenida_e-learning_curso.swf\) debemos dar hacer un clic derecho y escoger propiedades sobre el archivo para elegir la dirección real del archivo.](#)

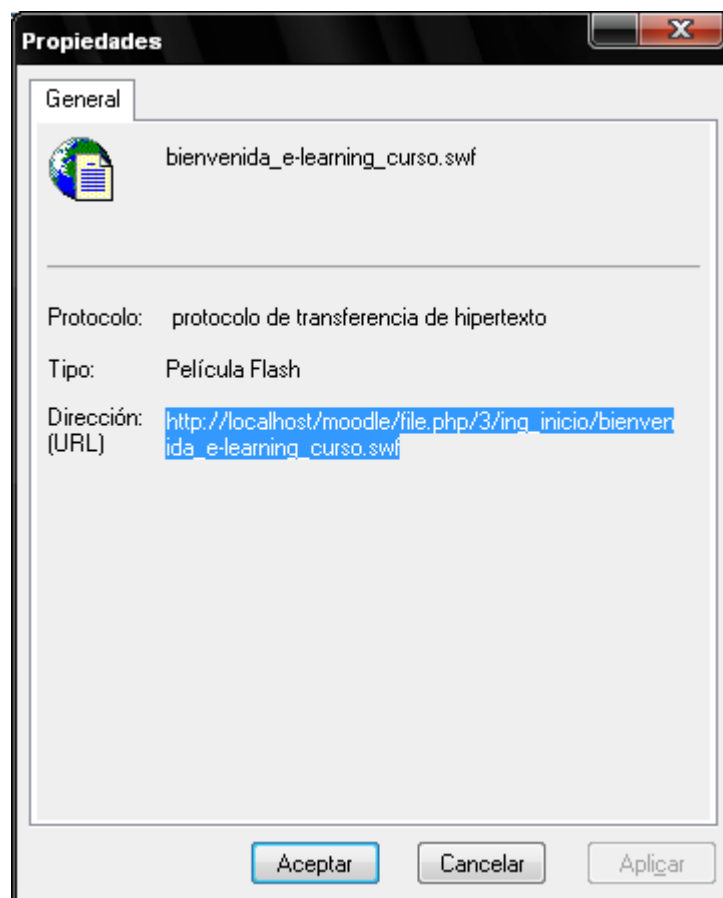


Figura 4.8 Dirección del archivo

Cuando ya hemos elegido el archivo y colocado la dirección, inmediatamente escogemos una opción que nos dice que el documento aparecerá en una misma pagina o en otra pagina. Y por ultimo hacemos clic en . Y finalmente aparecerá el documento en el sitio.

Para todos lo documentos que presente información, los paso que son los mismos.

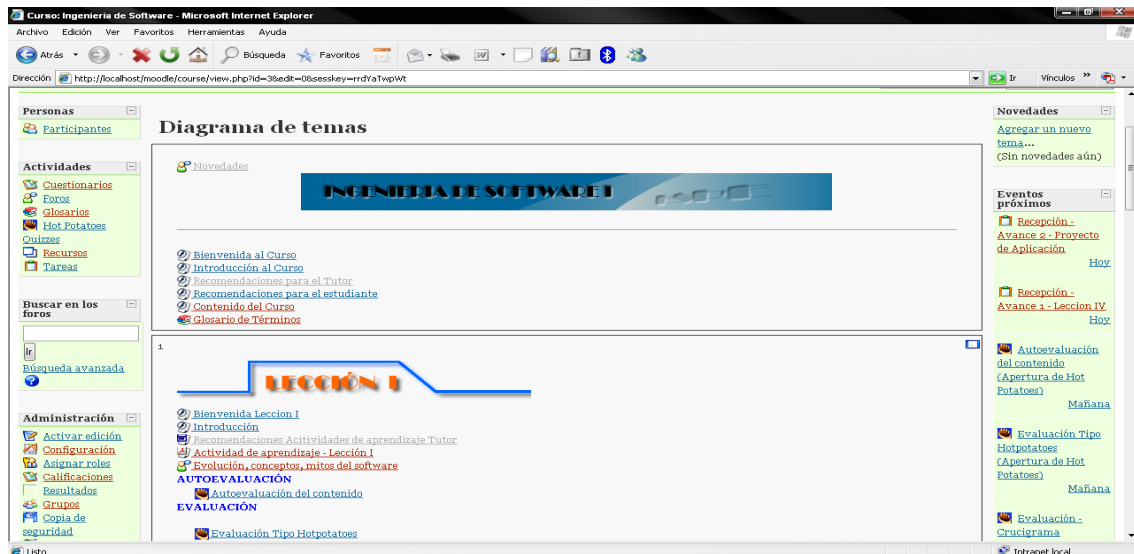


Figura 4.9 Dirección del archivo

4.7. Creación de cuestionarios

En nuestro proyecto se levanto dos tipos de evaluaciones que son: cuestionario que es una actividad que contiene Moodle y la actividad HotPotatoes que es una herramienta compatible con Moodle.

Cuestionario

Le ofrece la posibilidad de diseñar y plantear cuestionarios. Pueden ser: múltiple opción, falso/verdadero, respuestas cortas etc. Estos se conservan en la base de datos, lo que permite que sean reutilizados dentro del mismo curso o en otros.

Los cuestionarios posibilitan múltiples intentos. Cada uno se marca automáticamente y usted puede decidir dar a conocer o no la calificación y/o las respuestas correctas a los alumnos una vez concluido el cuestionario.

El examen también puede tener un feedback, donde se explique a los estudiantes el por qué de las respuestas. Para agregar “un cuestionario”, seleccione del menú “Agregar” la opción “Cuestionario”.

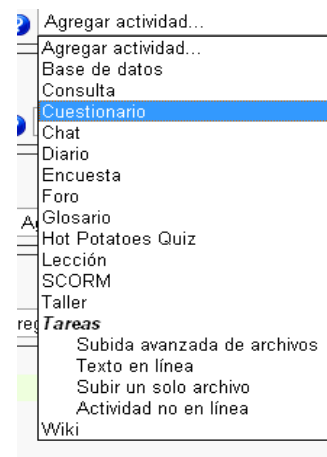


Figura 4.10 Cuestionario

Una vez que lo haya realizado, el sistema le abrirá la siguiente pantalla:

Agregando Cuestionario a tema 7

Ajustes generales

Nombre * Debe suministrar un valor aquí.

Introducción ?

Trebuchet 1 (8 pt) Idioma B I U S x₂ x² [Icons]

Ruta: ?

Tiempo

Abrir cuestionario ? 11 diciembre 2008 14 00 Deshabilitar

Cerrar cuestionario ? 11 diciembre 2008 14 00 Deshabilitar

Límite de tiempo (en minutos) ? 0 Habilitar

Tiempo entre el primer y el segundo intento ? Ninguno

Tiempo entre los intentos posteriores ? Ninguno

Mostrar

Figura 4.11 Cuestionario

Es imprescindible que complete los siguientes datos:

Nombre: título del examen. Este es el que aparecerá en la pantalla del campus.

Introducción: permite ingresar un párrafo introductorio al cuestionario que se verá una vez que elija esta opción.

Abrir cuestionario: especifica a partir de qué fecha está disponible el examen para los alumnos.

Cerrar cuestionario: indica hasta cuándo se podrá acceder al mismo.

Limite de tiempo.- en este indicamos por cuanto tiempo queremos que permanezca abierto el cuestionario.

Mezclar preguntas: si habilita esta opción, las preguntas del examen se ordenan aleatoriamente, cada vez que un estudiante intente resolverlo. La intención es evitar que se copien.

Mezclar respuestas: el sistema cambiará de lugar las respuestas del examen. Obviamente, esto sólo es aplicable a preguntas con varias opciones de respuesta, tales como las de múltiple opción o de emparejamiento.

Intentos permitidos: los estudiantes realizan múltiples intentos en un cuestionario. De esta manera, este instrumento, se transforma en una actividad educativa, brindando la posibilidad de aprender a partir del error.

Cada intento construye sobre el anterior: si selecciona la opción “Sí”, cada nuevo intento contendrá el resultado del anterior. Esto permite completar una pregunta a través de varios intentos. Para no mostrar la respuesta anterior en cada intento, seleccione “No”.

Método de calificación: cuando se realizan múltiples intentos, hay diferentes maneras en las que se puede calcular la puntuación final del examen:

- **Calificación más alta:** el puntaje final es el más alto en cualquiera de los intentos.
- **Puntuación media:** el resultado final es el promedio (la media simple) de la puntuación de todos los intentos.
- **Primera calificación:** sólo se tiene en cuenta la calificación final.
- **Última calificación:** la nota final es adquirida en el intento más reciente.

¿Mostrar resultados al finalizar?: si así lo desea, los estudiantes reciben la calificación total.

¿Mostrar la respuesta correcta al finalizar?: el alumno puede ver la respuesta correcta de cada pregunta (resaltada en un color brillante) una vez finalizado el cuestionario.

Permitir revisión: los estudiantes pueden revisar sus respuestas. Esto sólo es posible una vez que el cuestionario está cerrado.

Calificación máxima: la nota máxima colocada para un examen está en función del resto de calificaciones del curso.

Usted podría poner la calificación máxima en 20, porque el examen vale un 20% del curso entero. Así, aunque haya 10 preguntas en un examen, con un total de 50 puntos, las calificaciones de 50 puntos se reducirían a la calificación máxima de 20.

Continuar: una vez terminado de configurar el examen, se pasa a la siguiente pantalla:

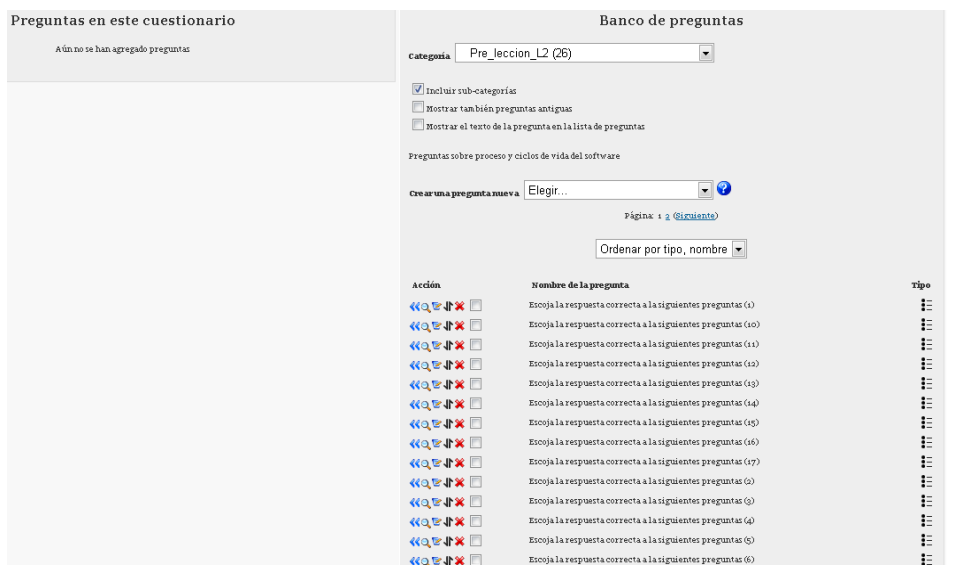


Figura 4.12 Preguntas de un cuestionario

Si observa en el extremo superior derecho de la pantalla, encontrará la opción “Categoría”, Allí podrá organizar las preguntas de acuerdo a su temática. Sólo necesita colocar un nombre y una breve descripción que las identifique. De esta manera, tendrá un material más ordenado y podrá acceder a él más rápidamente a la hora de confeccionar su examen.

Es de gran utilidad cuando su aula electrónica ha atravesado varios cuatrimestres y tiene muchas preguntas.

Para agregar categorías nuevas, pulse sobre “editar categorías” con lo que se verá una pantalla similar a la siguiente:



Figura 4.13 Categorías de una pregunta

Cuando ya elijamos la categoría seguiremos con la siguiente pantalla con la elección de las preguntas.

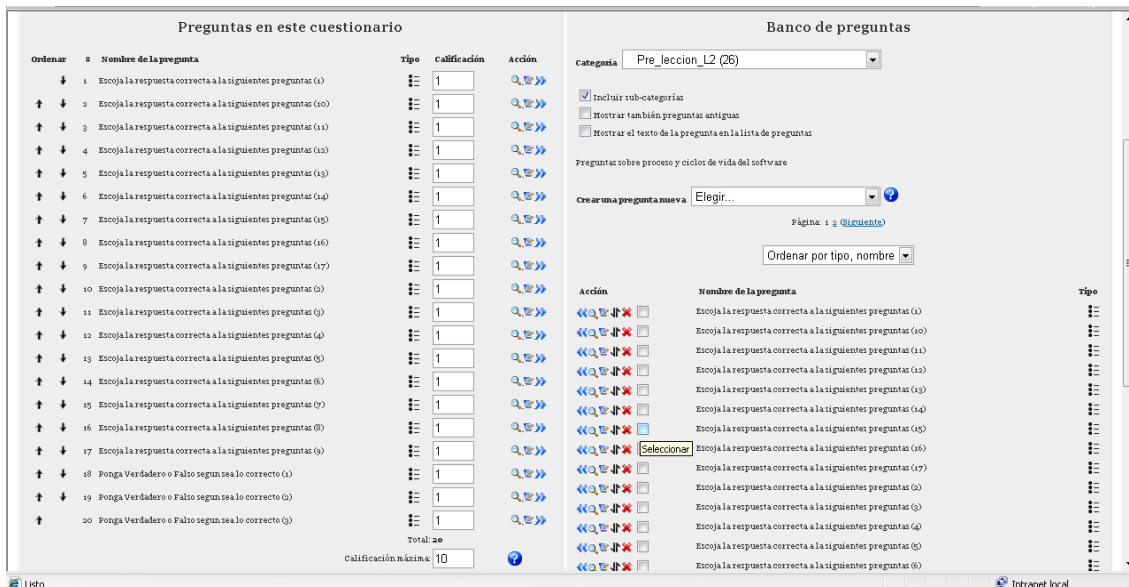


Figura 4.14 Categorías de una pregunta

Ahora, el cuestionario pasó a formar parte de su curso como lo muestra la siguiente figura:

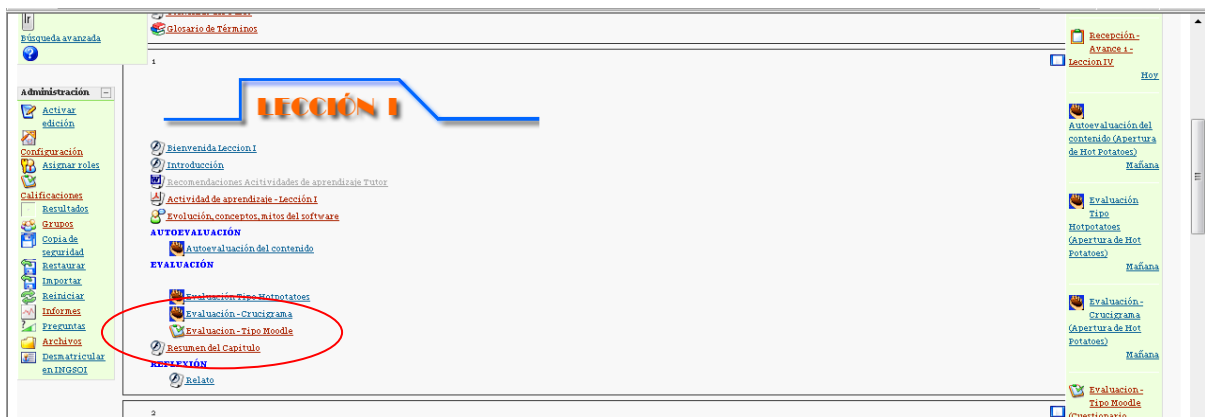


Figura 4.15 Cuestionario levantado en el curso

4.8. 🗣️ Foro

En nuestro proyecto se utiliza el foro como una actividad de aprendizaje para aumentar el conocimiento del alumno. Es aquí donde se desarrolla la mayor parte de los debates. Pueden estructurarse de diferentes maneras, e incluso permite evaluar la participación.

Los mensajes también se pueden ver de varias maneras, incluyendo mensajes e imágenes adjuntos. Al suscribirse a un foro, los participantes recibirán copias de cada mensaje en su buzón de correo electrónico. Usted puede imponer la suscripción a todos los integrantes del curso si así lo desea.

Es muy importante, ya que permite un mayor intercambio entre profesores y alumnos. Se puede estructurar de varias maneras e incluir calificaciones.

Los mensajes pueden presentarse con diferentes formatos, incluyendo archivos adjuntos. Al suscribirse a un foro, los participantes automáticamente reciben una copia de cada mensaje en su buzón de correo. Usted puede exigir a sus alumnos la suscripción al mismo.

Si así lo desea, puede crear un foro para dialogar con su clase.

Para agregar uno, seleccione del menú “Agregar” la opción “Foro”.

The screenshot shows a web interface for configuring a forum. At the top, it says "Actualizando Foro en tema 1". The main section is titled "Ajustes generales" and includes the following fields:

- Nombre del foro:** "Evolución, conceptos, mitos del software"
- Tipo de foro:** "Foro para uso general"
- Introducción:** A rich text editor containing the text: "Entre los períodos de la informática comprendido entre 1960 - 1990, el principal desafío era el desarrollo del hardware de computadoras, de forma que se redujera el costo del procesamiento y almacenamiento de datos, etc. A lo largo de las décadas los avances en microelectrónica han dado como resultado una mayor potencia de cálculo a la vez que una reducción del costo. Sin embargo".
- Ruta:** A field with a question mark icon.
- Forzar la suscripción de todos:** "No"
- Leer rastro de este foro:** "Opcional"
- Tamaño máximo del archivo adjunto:** "2Mb"

The bottom section is titled "calificación" and includes:

- Permitir la calificación de los mensajes:** "Usar calificaciones" (checked)
- Calificación:** "10"
- Restringir las calificaciones a mensajes colocados en este período:** (checked)
- Desde:** "3 de diciembre de 2008 17:00"

Figura 4.16 Foro

Nombre del foro: nombre que se va a presentar en la pantalla de los alumnos.

Tipo de foro: ofrece tres opciones:

- **Cada persona plantea un tema:** esta modalidad es conveniente cuando usted propone como actividad que cada estudiante empiece una discusión y que todos los demás le respondan.
- **Debate sencillo:** los estudiantes pueden contestar el tema, pero no crear nuevos. Simplemente, es un intercambio de ideas sobre un solo tema. Es útil para debates cortos y muy concretos.
- **Foro para uso general:** es un foro abierto donde cualquiera puede empezar un nuevo tema de debate cuando desee. Es el más adecuado para uso general.

Introducción: en este campo, debe realizar una descripción del tema central a tratarse en el foro.

Permitir que cualquier estudiante abra un nuevo tema: esta opción tiene las siguientes opciones:

- **Permitir nuevos temas y respuestas:** aquí, brinda la posibilidad de incorporar nuevos comentarios a los debates ya contestados.
- **No se puede colocar temas, sólo respuestas:** sólo usted puede iniciar nuevos debates; los estudiantes se limitan a responder.
- **No se pueden colocar temas ni respuestas:** es el espacio donde usted anuncia los nuevos asuntos que aparecen en la página principal del curso.

Aparentemente, existen contradicciones entre lo que se explica en “Tipo de foro” y “Permitir que cualquier estudiante abra un nuevo tema”. Para aclararlas, se presenta la siguiente tabla:

Tabla 4.1 Descripción del foro

ESTUDIANTE	Foro para uso general	Cada persona plantea un tema	Debate sencillo
Permitir nuevos temas y respuestas	Permite crear varios temas y responder a todos los creados	Se puede incorporar un solo tema y responder a los creados por los demás	No se puede crear temas, sólo responder
No se puede colocar temas solo respuestas	No se puede crear temas, sólo se puede responder	Se puede crear un solo tema y responder a los creados por los demás	No se puede crear temas, sólo responder
No se puede colocar temas ni respuestas	No se puede crear temas ni responder a los mismos	Se puede crear un solo tema y responder a los creados por los demás	No se permite crear temas ni responderlos

Permitir la calificación de los mensajes: todo mensaje enviado puede ser calificado de acuerdo a la calidad de la respuesta dada.

Existen varias escalas, entre ellas tenemos las siguientes:

- **Vías de conocimiento separadas y conectadas:** son las basadas en la teoría del **conocimiento individualista y relacionado**. Las mismas, le permiten observar las interacciones humanas desde un nuevo punto de vista.

Entendemos por conocimiento individualista a aquel que es adquirido sin intervención de otros.

A Juan le agrada permanecer tan “objetivo” como le sea posible, sin que influyan los sentimientos y emociones en su aprendizaje. Cuando entra en debate con otras personas que tienen ideas diferentes, le gusta defender las propias, encontrando puntos débiles en los argumentos de sus oponentes. Es crítico con las nuevas ideas, a menos que sean hechos probados de fuentes refutadas, como libros de texto, maestros respetados o de la experiencia directa. Prefiere trabajar solo y sacar sus propias conclusiones.

Por el contrario, conocimiento relacionado, es aquel que se adquiere en conexión con los pares.

María es experta en empatía y tiende a escuchar y hacer preguntas hasta que siente que puede aclarar sus dudas. Aprende escuchando las experiencias de otros. Al hablar, evita la confrontación e intenta, a menudo, ayudar a la otra persona si encuentra la manera de hacerlo, mediante sugerencias lógicas.

Para obtener un aprendizaje colaborativo y eficaz, lo ideal es que todos utilicen **ambas** maneras de aprender.

Si usted presta atención, cuando un alumno envía un mensaje al foro, refleja en él sus actitudes frente al aprendizaje.


Al utilizar esta escala para evaluarlo, podrá lograr:

- a) Ayudarle a pensar sobre estos asuntos al leer otros mensajes.
- b) Proporcionar retroalimentación a cada autor acerca de cómo ellos están siendo vistos por los demás.

- **Escala numérica:** evaluada de 0 a 100.

Forzar la inscripción a todos: cuando un estudiante se suscribe a un foro, usted debe remitirle por correo electrónico una copia de cada mensaje enviado (éstos son despachados aproximadamente 30 minutos después de haber sido escritos).

Los alumnos, normalmente, pueden escoger si desean o no suscribirse. Sin embargo, si usted desea que todos participen en un foro concreto, esta posibilidad de elección es anulada y cada uno recibirá las copias de los mensajes por correo electrónico. Esto es especialmente útil en el foro de las noticias y en los iniciados al principio del curso.

Cuando usted haya culminado, haga clic sobre “Guardar cambios”, y verá el icono de foro  en la página del curso. (Ver figura 4.17).

Si desea trabajar con esta opción, sólo debe colocarse sobre dicho link.



Figura 4.17 Foro en el curso

4.9. Glosario

Este recurso permite la creación de un glosario de términos, generado ya sea por los profesores o por los estudiantes.

Cada definición puede ir acompañada o no del nombre del usuario que la creó. Realiza búsquedas por inicial, por término, o por caracteres especiales. Para ingresar, elija del menú “Agregar”, la opción “Glosario”. Ver figura 4.18.

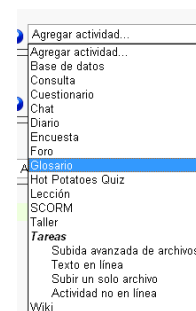


Figura 4.18 Glosario

Luego, percibirá una pantalla como la siguiente:

Figura 4.19 Creación de un Glosario

Nombre: nombre que le da al glosario.

Descripción: breve descripción del mismo.

Entradas por páginas: especifica el número de entradas que serán mostradas en cada página.

Tipo de glosario: el sistema le permite exportar las entradas de cualquier glosario secundario al principal del curso. Para hacer esto, debe especificar cuál es el principal.

Nota: usted puede tener sólo un glosario principal por curso y únicamente los docentes tienen la posibilidad de actualizarlo.

Los estudiantes pueden ingresar entradas: determina si le va a permitir o no a un estudiante agregar, revisar o anular sus propias entradas. Las exportadas a un glosario principal, únicamente pueden ser actualizadas o anuladas por usted. Esto sólo se aplica a los glosarios secundarios.

Nota: usted puede revisar o anular cualquier entrada en el momento que desee.

Permitir entradas duplicadas: si habilita esta opción, un mismo concepto podrá tener múltiples entradas.

Permitir comentar las entradas: si así lo desea, puede permitir que otros agreguen comentarios a las entradas del glosario.

Hiperenlaces automático: al activarla, posibilita que las entradas en este glosario sean ligadas automáticamente, cada vez que las palabras y las frases aparezcan dentro del mismo curso.

Estado de aprobación por defecto: elige si las entradas del estudiante se incorporan automáticamente o si éstas deben ser previamente aprobadas por usted.

Formato de muestra de entrada: esta variable define la manera cómo se mostrará cada entrada dentro del glosario. Los formatos disponibles son:

- **Diccionario simple:** luce como un diccionario convencional, con las entradas separadas. No se muestra ningún autor y los archivos adjuntos aparecen como enlaces.
- **Continuo:** presenta las entradas una después de la otra sin otra separación que los iconos.
- **Completo con autor:** posee formato semejante al de los foros, que muestra la información del autor de cada entrada. Los archivos adjuntos aparecen como enlaces.
- **Completo sin autor:** el formato es semejante al de los foros. No muestra la información del autor de cada entrada. Los archivos adjuntos aparecen como enlaces.
- **Enciclopedia:** su formato es semejante a “Completo con autor”, con la diferencia, que aquí las imágenes adjuntas se muestran en línea.
- **FAQ:** útil para presentar listas de “Preguntas Frecuentemente Respondidas”. Automáticamente, añade las palabras “Pregunta y Respuesta” en el concepto y la definición respectivamente.

Mostrar enlace “Especial”: habilita la búsqueda con caracteres especiales como @, #, etc.

Mostrar alfabeto: permite la exploración por letras iniciales.

Mostrar enlace “TODAS”: habilita o no todas las características de una vez.

Permitir calificar las entradas: se puede optar por:

- Usar calificaciones
- Usuarios:
 - Solamente los profesores clasifican las entradas
 - Todos pueden clasificarlas.

- **Calificación:** le permite seleccionar distintas escalas de calificación.
- **Restringir las calificaciones a las entradas** cuando las fechas estén en estos rangos. Aquí puede poner fechas a las calificaciones.

Finalmente, con “Guardar cambios”, se guarda el glosario y queda disponible para ser utilizado.

En la parte superior de la pantalla, se encuentra un botón “Buscar” que le permite realizar búsquedas de términos en el glosario, y a la derecha del mismo, un pequeño recuadro que, si usted lo marca, buscará dichas palabras en los conceptos y las definiciones.

En esta pantalla, se encuentran dos líneas de links que nos permiten realizar distintas actividades. Veámoslas detalladamente:

Agregar entradas: posibilita añadir un nuevo vocablo y su significado al glosario. Al elegir esta opción, el sistema lo lleva a una nueva pantalla que se muestra en la figura 4.18



Figura 4.20 Glosario de Términos

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Conclusiones Generales

- La educación E-learning sin duda puede aplicarse para la educación técnica, solo que se debe tomar en cuenta, con las herramientas o tecnología adecuada para obtener un aprovechamiento óptimo.
- Esta nueva tecnología con el pasar del tiempo no reemplazará al aula si no que simplemente la educación se trasladará a nuestro lugar de trabajo u hogar.

Conclusiones sobre el Desarrollo del Curso E-learning

- Las actividades de aprendizaje actualmente cuenta con formas cada vez más interactivas para el usuario lo que nos permite medir y valorar cualitativamente al usuario y que facilitan la tarea a los tutores.
- Las actividades de aprendizaje se deben aplicar en base a experiencias reales para que el estudiante tenga la capacidad de reconocer y analizar la aplicación que tienen la carrera que está siguiendo.
- El método Ergológico nos dio las pautas que permitieron realizar este curso e-Learning de una manera ordenada siguiendo sus etapas de análisis, diseño e implementación.

- La plataforma Moodle permite tener un sistema de enseñanza cada vez más interactivo, y con la utilización de herramientas Macromedia, nuestro curso tuvo la característica de tener audio en las partes relevantes de cada lección, lo que nos abre una puerta para mejorar las unidades de las asignaturas y de esta manera expandir el campo de enseñanza.
- Los estudiantes que tomen el curso de e-learning de Ingeniería de Software I podrán desarrollar destrezas referentes a la asignatura, debido que en aquellos temas en los cuales su contenido es basado en estimaciones y fórmulas, ejercicios ejemplo mejorarán su comprensión.
- De los foros se concluyó que es una manera en la cual los alumnos pueden despejar sus dudas en algún tema, y el tutor en base a los resultados deberá tomar decisiones para evitar dejar vacíos en los contenidos.

5.2. RECOMENDACIONES

Se recomienda que:

- Para que la plataforma de educación virtual tenga éxito en la ESPE se debe seguir desarrollando cursos y aplicar de manera paulatina con los estudiantes con el fin de encontrar y mejorar errores como por ejemplo la planificación de un curso.
- La utilización de la plataforma Moodle es una gran ventaja, ya que esta ofrece una gran cantidad de herramientas y como a manera de recomendación se sugiere que en las próximas tesis se investigue y aplique una herramienta que esta en auge como lo es la video conferencia en los cursos.
- Para realizar la implementación de cursos en e-learning, se debe basar en metodologías que guíen a obtener una óptima comprensión del alumno y permitan alcanzar la meta al concluir el curso.

- En vista que es un curso de e-learning, el contenido del curso debe ser lo más entendible posible para el alumno, para que las dudas con respecto a un tema sea resuelto con el material expuesto en el curso y no se requiera de tutorías diarias.
- Se recomienda que el Departamento de Ciencias de la Computación de un mayor empuje a esta iniciativa de tener la modalidad de e-learning, como inicio considerarlo como material de apoyo, para que posteriormente cierto tipo de materias sean presenciales y otras no, de esta manera también se apoyaría aquellas personas que por horarios de trabajo o situación geográfica no pueden asistir a clases.
- En vista que la plataforma Moodle da la apertura de añadir cualquier tipo de archivo, sea este en Word, Pdf, documentos flash, etc; se recomienda que los cursos sean realizados según las exigencias actuales de los alumnos y a la vez sean amigables para incentivarlos a tomar el curso e-learning.