



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

**ESTRUCTURACIÓN DE LOS LABORATORIOS Y DOCUMENTACIÓN DE APOYO  
DE LOS CURSOS: ANÁLISIS Y DISEÑO DE SISTEMAS 1, ANÁLISIS Y DISEÑO DE  
SISTEMAS 2, SOFTWARE AVANZADO APLICANDO LA METODOLOGÍA DEL IT  
CENTER INDIA-GUATEMALA, DE LA CARRERA DE INGENIERÍA EN CIENCIAS Y  
SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE  
SAN CARLOS DE GUATEMALA**

**Heber Arnulfo Pernilla Álvarez**

Asesorado por el Ingeniero Jorge Armín Mazariegos

Guatemala, noviembre de 2009



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ESTRUCTURACIÓN DE LOS LABORATORIOS Y DOCUMENTACIÓN DE APOYO DE LOS CURSOS: ANÁLISIS Y DISEÑO DE SISTEMAS 1, ANÁLISIS Y DISEÑO DE SISTEMAS 2, SOFTWARE AVANZADO APLICANDO LA METODOLOGÍA DEL IT CENTER INDIA-GUATEMALA, DE LA CARRERA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR:

**HEBER ARNULFO PERNILLA ÁLVAREZ**

ASESORADO POR EL ING. JORGE ARMIN MAZARIEGOS

AL CONFERÍRSELE EL TÍTULO DE  
**INGENIERO EN CIENCIAS Y SISTEMAS**  
GUATEMALA, NOVIEMBRE DE 2009



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soría
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. José Milton de León Bran
VOCAL V	Br. Isaac Sultan Mejía
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Inga. Floriza Felipa Ávila Pesquera de Mendinilla
EXAMINADOR	Ing. Marlon Antonio Pérez Türk
EXAMINADOR	Inga. Sonia Yolanda Castañeda Ramírez
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas



## **HONORABLE TRIBUNAL EXAMINADOR**

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

**ESTRUCTURACIÓN DE LOS LABORATORIOS Y DOCUMENTACIÓN DE APOYO DE LOS CURSOS: ANÁLISIS Y DISEÑO DE SISTEMAS 1, ANÁLISIS Y DISEÑO DE SISTEMAS 2, SOFTWARE AVANZADO APLICANDO LA METODOLOGÍA DEL IT CENTER INDIA-GUATEMALA, DE LA CARRERA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA,**

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, en septiembre 2009.

Heber Arnulfo Pernilla Álvarez





Guatemala, 10 de Octubre 2009

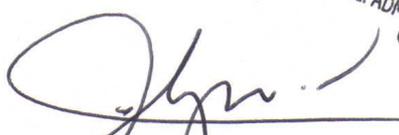
Ingeniera  
Norma Ileana Sarmiento  
Directora Unidad EPS  
Dirección de EPS  
Facultad de Ingeniería  
USAC

Respetable Ingeniera Sarmiento:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **HEBER ARNULFO PERNILLA ÁLVAREZ**, titulado: **“ESTRUCTURACIÓN DE LOS LABORATORIOS Y DOCUMENTACIÓN DE APOYO DE LOS CURSOS: ANÁLISIS Y DISEÑO DE SISTEMAS 1, ANÁLISIS Y DISEÑO DE SISTEMAS 2, SOFTWARE AVANZADO APLICANDO LA METODOLOGÍA DEL IT CENTER INDIA-GUATEMALA, DE LA CARRERA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, y a mi criterio, el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Agradeciendo su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
Ing. Jorge Armin Mazariegos  
Asesor del Trabajo de Graduación  
Ingeniería en Ciencias y Sistemas

Jorge Armin Mazariegos Rabanales  
ING. EN CIENCIAS Y SISTEMAS  
M.C. ADMINISTRADOR DE TECNOLOGIA  
COLEGIADO 5547



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería



UNIDAD DE E.P.S.

Guatemala, 15 de octubre de 2009.  
Ref.EPS.DOC.1456.10.09.

Inga. Norma Ileana Sarmiento Zeceña de Serrano  
Directora Unidad de EPS  
Facultad de Ingeniería  
Presente

Estimada Ingeniera Sarmiento Zeceña.

Por este medio atentamente le informo que como Supervisora de la Práctica del Ejercicio Profesional Supervisado, (E.P.S) del estudiante universitario de la Carrera de Ingeniería en Ciencias y Sistemas, **Heber Arnulfo Pernilla Álvarez** Carné No. **200313540** procedí a revisar el informe final, cuyo título es **“ESTRUCTURACIÓN DE LABORATORIOS Y DOCUMENTACIÓN DE APOYO DE LOS CURSOS: ANÁLISIS Y DISEÑO DE SISTEMAS 1, ANÁLISIS Y DISEÑO DE SISTEMAS 2 Y SOFTWARE AVANZADO APLICANDO LA METODOLOGÍA DEL IT CENTER INDIA-GUATEMALA, DE LA CARRERA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**.

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

*“Id y Enseñad a Todos”*

Inga. Floriza Felipa Avila Pesquera de Medinilla

Supervisora de EPS

Área de Ingeniería en Ciencias y Sistemas

FFAPdM/RA







UNIDAD DE E.P.S.

Guatemala, 15 de octubre de 2009.  
Ref.EPS.D.707.10.09.

Ing. Marlon Antonio Pérez Turck  
Director Escuela de Ingeniería Ciencias y Sistemas  
Facultad de Ingeniería  
Presente

Estimado Ingeniero Perez Turck.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **“ESTRUCTURACIÓN DE LABORATORIOS Y DOCUMENTACIÓN DE APOYO DE LOS CURSOS: ANÁLISIS Y DISEÑO DE SISTEMAS 1, ANÁLISIS Y DISEÑO DE SISTEMAS 2 Y SOFTWARE AVANZADO APLICANDO LA METODOLOGÍA DEL IT CENTER INDIA-GUATEMALA, DE LA CARRERA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, que fue desarrollado por el estudiante universitario **Heber Arnulfo Pernilla Álvarez** Carné No. **200313540** quien fue debidamente asesorado por el Ing. Jorge Armin Mazariegos y supervisado por la Inga. **Floriza Felipa Ávila Pesquera de Medinilla**

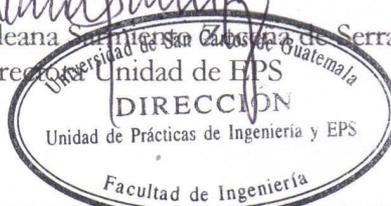
Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor y de la Supervisora de EPS, en mi calidad de Directora apruebo su contenido solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

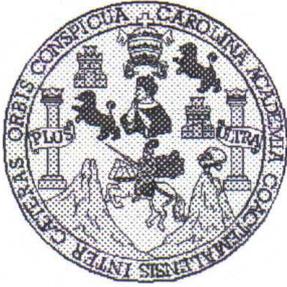
*“Id y Enseñad a Todos”*

Inga. Norma Ileana Serrano de Serrano  
Directora Unidad de EPS



NISZ/ra





Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 21 de Octubre de 2009

Ingeniero  
**Marlon Antonio Pérez Turk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **HEBER ARNULFO PERNILLA ALVAREZ**, titulado: **"ESTRUCTURACIÓN DE LOS LABORATORIOS Y DOCUMENTACIÓN DE APOYO DE LOS CURSOS: ANALISIS Y DISEÑO DE SISTEMAS 1, ANALISIS Y DISEÑO DE SISTEMAS 2, SOFTWARE AVANZADO APLICANDO LA METODOLOGIA DEL IT CENTER INDIA-GUATEMALA, DE LA CARRERA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación





E  
S  
C  
U  
E  
L  
A  
  
D  
E  
  
C  
I  
E  
N  
C  
I  
A  
S  
  
Y  
  
S  
I  
S  
T  
E  
M  
A  
S

UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado **“ESTRUCTURACIÓN DE LOS LABORATORIOS Y DOCUMENTACIÓN DE APOYO DE LOS CURSOS: ANÁLISIS Y DISEÑO DE SISTEMAS 1, ANÁLISIS Y DISEÑO DE SISTEMAS 2, SOFTWARE AVANZADO APLICANDO LA METODOLOGÍA DEL IT CENTER INDIA-GUATEMALA, DE LA CARRERA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, presentado por el estudiante **HEBER ARNULFO PERNILLA ÁLVAREZ**, aprueba el presente trabajo y solicita la autorización del mismo.*

**“ID Y ENSEÑAD A TODOS”**

*Ing. Marlon Antonio Pérez Turk*

Director, Escuela de Ingeniería en Ciencias y Sistemas



Guatemala, 12 de noviembre 2009





El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **ESTRUCTURACIÓN DE LOS LABORATORIOS Y DOCUMENTACIÓN DE APOYO DE LOS CURSOS: ANÁLISIS Y DISEÑO DE SISTEMAS 1, ANÁLISIS Y DISEÑO DE SISTEMAS 2, SOFTWARE AVANZADO APLICANDO LA METODOLOGÍA DEL IT CENTER INDIA-GUATEMALA, DE LA CARRERA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por el estudiante universitario **Heber Arnulfo Pernilla Alvarez**, autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Murphy Olympo Paiz Recinos  
DECANO



Guatemala, noviembre de 2009

/cc  
c.c. archivo.



## **ACTO QUE DEDICO A:**

Dios: La fortaleza de mi vida, y fuente de mi inspiración, quien me sostuvo hasta el final de mi carrera y ahora me permite culminar esta etapa de mi vida con éxito.

Mis padres: A mis padres, Arnulfo Pernilla e Izabel de Pernilla, quienes me han apoyado durante toda mi vida, en los momentos alegres y durante la adversidad.



## **AGRADECIMIENTOS A:**

Universidad San Carlos de Guatemala	Por haber mantenido durante toda mi carrera un alto nivel de educación.
Mi asesor	Ing. Armin Mazariegos, por el tiempo dedicado y los consejos aportados al proyecto.
Director de Escuela	Ing. Marlon Pérez quien me apoyó en las gestiones diversas de la Escuela de Ingeniería en Ciencias y Sistemas para realizar el presente proyecto de graduación.
Inga. Floriza Ávila	Por su asesoría en los procesos de práctica laboral y desarrollo de EPS.
Mis hermanos	Daniel y Brenda, quienes me apoyaron a lo largo de mi carrera.
Mi tía Eulalia Álvarez	Por el apoyo brindado a lo largo de mi vida.
Colaboradores	A todas las personas que hicieron posible la realización de este proyecto, al proveer sus consejos y aportaciones al proyecto.



# ÍNDICE GENERAL

<b>ÍNDICE DE ILUSTRACIONES.....</b>	<b>V</b>
<b>GLOSARIO.....</b>	<b>XI</b>
<b>RESUMEN.....</b>	<b>XIII</b>
<b>OBJETIVOS .....</b>	<b>XV</b>
<b>INTRODUCCIÓN.....</b>	<b>XVII</b>
<b>1. MARCO TEÓRICO .....</b>	<b>1</b>
1.1 Principios de pedagogía .....	1
1.2 Proceso de enseñanza – aprendizaje.....	4
1.3 Método de clases magistrales .....	6
1.4 Aprendizaje práctico por medio de laboratorios.....	8
<b>2. DEFINICIÓN DE UNA GUÍA DEL INSTRUCTOR .....</b>	<b>11</b>
2.1 Objetivos de una guía de instructor .....	13
2.2 Elementos de una guía de instructor .....	13
2.3 Proceso de construcción de la guía del instructor .....	14
2.4 Forma de uso de la guía de instructor .....	20
<b>3. GUÍA DEL INSTRUCTOR DE ANÁLISIS Y DISEÑO DE SISTEMAS 1 .....</b>	<b>21</b>
3.1 Información general .....	21
3.2 Distribución de horas y actividades .....	22
3.3 Evaluación .....	23
3.4 Detalle de sesiones .....	24
3.5 Distribución de sesiones.....	25
3.6 Detalle de tareas asignadas .....	38
3.7 Detalle de tutoriales .....	40
3.8 Detalle de evaluaciones.....	41

3.9	Evaluación de ejemplo .....	42
3.10	Bibliografía recomendada .....	44
3.11	Anexos .....	46
<b>4.</b>	<b>GUÍA DEL INSTRUCTOR DE ANÁLISIS Y DISEÑO DE SISTEMAS 2 .....</b>	<b>53</b>
4.1	Información general.....	53
4.2	Distribución de horas y actividades .....	54
4.3	Evaluación.....	54
4.4	Detalle de sesiones .....	55
4.5	Distribución de sesiones.....	56
4.6	Detalle de tareas asignadas .....	67
4.7	Detalle de tutoriales.....	67
4.8	Detalle de evaluaciones .....	69
4.9	Evaluación de ejemplo .....	70
4.10	Bibliografía recomendada .....	71
4.11	Anexos .....	72
<b>5.</b>	<b>GUÍA DEL INSTRUCTOR DE SOFTWARE AVANZADO .....</b>	<b>77</b>
5.1	Información general.....	77
5.2	Distribución de horas y actividades .....	78
5.3	Evaluación.....	78
5.4	Detalle de sesiones .....	79
5.5	Distribución de sesiones.....	80
5.6	Detalle de tareas asignadas .....	91
5.7	Detalle de tutoriales.....	92
5.8	Detalle de evaluaciones .....	95
5.9	Evaluación de ejemplo .....	96
5.10	Bibliografía recomendada .....	98
5.11	Anexos .....	99
<b>6.</b>	<b>DOCUMENTACIÓN DE APOYO DE “ANÁLISIS Y DISEÑO DE SISTEMAS 1” .....</b>	<b>101</b>

6.1	Descripción general .....	101
6.2	Ingeniería de software .....	103
6.3	Métodos de desarrollo de software .....	104
6.4	Administración de procesos de negocio .....	108
6.5	Administración de requerimientos con RUP .....	110
6.6	Administración de requerimientos con Extreme Programming .....	111
6.7	Usabilidad .....	113
6.8	Demo: capítulo 1 “Ingeniería de Software” .....	115
<b>7.</b>	<b>DOCUMENTACIÓN DE APOYO DE “ANÁLISIS Y DISEÑO DE SISTEMAS 2” .....</b>	<b>119</b>
7.1	Descripción general .....	119
7.2	Administración de la configuración .....	121
7.3	Integración continua .....	122
7.4	Arquitectura de software .....	124
7.5	Arquitectura orientada a servicios.....	127
7.6	Análisis de clases .....	129
7.7	Patrones de diseño .....	130
7.8	Demo: capítulo 1 “Administración de la Configuración” .....	133
<b>8.</b>	<b>DOCUMENTACIÓN DE APOYO DE “SOFTWARE AVANZADO” .....</b>	<b>143</b>
8.1	Descripción general .....	143
8.2	Microsoft Solutions Framework.....	145
8.3	Modelo de equipo .....	146
8.4	Modelo de proceso .....	147
8.5	Disciplinas de gestión de proyectos y gestión de riesgos .....	148
8.6	Métricas de proyectos.....	150
8.7	Costo total de propiedad y retorno de inversión .....	151
8.8	Demo: capítulo 1 “Microsoft Solutions Framework” .....	152
	<b>CONCLUSIONES .....</b>	<b>159</b>
	<b>RECOMENDACIONES.....</b>	<b>161</b>

**BIBLIOGRAFÍA..... 163**  
**ANEXOS ..... 167**

# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1. Portada Libro Análisis y Diseño de Sistemas 1.....	102
2. Páginas 13-16 Libro “Análisis y Diseño de Sistemas 1”.....	115
3. Páginas 17-20 Libro “Análisis y Diseño de Sistemas 1”.....	116
4. Páginas 21-24 Libro “Análisis y Diseño de Sistemas 1”.....	117
5. Página 25 Libro “Análisis y Diseño de Sistemas 1”.....	118
6. Portada del Libro de Análisis y Diseño de Sistemas 2.....	120
7. Páginas 18-21 Libro “Análisis y Diseño de Sistemas 2”.....	133
8. Páginas 22-25 Libro “Análisis y Diseño de Sistemas 2”.....	134
9. Páginas 26-29 Libro “Análisis y Diseño de Sistemas 2”.....	135
10. Páginas 30-33 Libro “Análisis y Diseño de Sistemas 2”.....	136
11. Páginas 34-37 Libro “Análisis y Diseño de Sistemas 2”.....	137
12. Páginas 38-41 Libro “Análisis y Diseño de Sistemas 2”.....	138
13. Páginas 42-45 Libro “Análisis y Diseño de Sistemas 2”.....	139
14. Páginas 46-49 Libro “Análisis y Diseño de Sistemas 2”.....	140
15. Páginas 50-53 Libro “Análisis y Diseño de Sistemas 2”.....	141
16. Páginas 54-57 Libro “Análisis y Diseño de Sistemas 2”.....	142
17. Portada del Libro de <i>Software Avanzado</i> .....	144
18. Páginas 10-13 Libro “ <i>Software Avanzado</i> ”.....	152
19. Páginas 14-17 Libro “ <i>Software Avanzado</i> ”.....	153
20. Páginas 18-21 Libro “ <i>Software Avanzado</i> ”.....	154

21. Páginas 22-25 Libro “Software Avanzado” .....	155
22. Páginas 26-29 Libro “Software Avanzado” .....	156
23. Páginas 30-31 Libro “Software Avanzado” .....	157

## TABLAS

I. Distribución de horas y actividades.....	15
II. Forma de evaluación .....	16
III. Detalle de sesiones.....	16
IV. Distribución del curso Análisis y Diseño de Sistemas 1 .....	22
V. Evaluación del curso de Análisis y Diseño de Sistemas 1 .....	23
VI. Detalle de sesiones del curso de Análisis y Diseño de Sistemas 1 .....	24
VII. Sesión 1: conceptos de sistemas de información .....	25
VIII. Sesión 2: proceso de desarrollo de software .....	26
IX. Sesión 3: metodologías de desarrollo de <i>software</i> .....	26
X. Sesión 4: metodologías rígidas.....	27
XI. Sesión 5: metodologías ágiles .....	27
XII. Sesión 6: <i>Extreme Programming</i> .....	28
XIII. Sesión 7: <i>Dynamics System Development Method</i> .....	29
XIV. Sesión 8: <i>Open Source Software</i> .....	30
XV. Sesión 9: presentaciones: “metodologías de desarrollo” .....	30
XVI. Sesión 10: casos de uso.....	31
XVII. Sesión 11: ejercicios de casos de uso .....	32
XVIII. Sesión 12: requerimientos no funcionales .....	32
XIX. Sesión 13: usabilidad.....	33
XX. Sesión 14: historias de usuario.....	33
XXI. Sesión 15: ejercicios de historias de usuario .....	34
XXII. Sesión 16: disciplina de pruebas de RUP .....	34

XXIII.	Sesión 17: modelo CMMI .....	35
XXIV.	Sesión 18: ITIL .....	36
XXV.	Sesión 19: presentaciones proceso de pruebas de <i>software</i> .....	36
XXVI.	Sesión 20: sesión extra .....	37
XXVII.	Sesión 21: evaluación final.....	37
XXVIII.	Asignación de Laboratorio 1.....	38
XXIX.	Asignación de Laboratorio 2.....	39
XXX.	Tutorial 1 .....	40
XXXI.	Tutorial 2 .....	41
XXXII.	Evaluaciones.....	41
XXXIII.	Cuadro comparativo “modelos del ciclo de desarrollo” (tutorial de ejemplo) .....	48
XXXIV.	Cuadro comparativo “metodologías de desarrollo” (tutorial de ejemplo) .....	49
XXXV.	Distribución del curso Análisis y Diseño de Sistemas 2 .....	54
XXXVI.	Evaluación del curso de Análisis y Diseño de Sistemas 2 .....	55
XXXVII.	Detalle de sesiones del curso de Análisis y Diseño de Sistemas 2.....	55
XXXVIII.	Sesión 1: administración de la configuración .....	56
XXXIX.	Sesión 2: versionamiento de sistemas de información.....	57
XL.	Sesión 3: plan de la administración de la configuración.....	57
XLI.	Sesión 4: requerimientos no funcionales.....	58
XLII.	Sesión 5: presentaciones de estudiantes: “herramientas para administración de la configuración” .....	58
XLIII.	Sesión 6: configuración de una herramienta para administración de la configuración .....	59
XLIV.	Sesión 7: <i>SPRING</i> .....	59
XLV.	Sesión 8: mapeo objeto relacional .....	60
XLVI.	Sesión 9: implementación de ORM en diversas tecnologías .....	61
XLVII.	Sesión 10: modelo vista controlador .....	62

XLVIII.	Sesión 11: arquitectura de <i>software</i> .....	62
XLIX.	Sesión 12: servidor de aplicaciones.....	63
L.	Sesión 13: componentes en servidores de aplicaciones .....	63
LI.	Sesión 14: pruebas de <i>software</i> .....	64
LII.	Sesión 15: presentaciones: “herramientas para pruebas” .....	64
LIII.	Sesión 16: integración de sistemas .....	65
LIV.	Sesión 17: <i>Message Broker</i> .....	65
LV.	Sesión 18: herramienta <i>Message Broker</i> .....	66
LVI.	Sesión 19: sesión extra.....	66
LVII.	Sesión 20: evaluación final .....	67
LVIII.	Tutorial 1 .....	67
LIX.	Tutorial 2.....	68
LX.	Tutorial 3.....	68
LXI.	Evaluaciones .....	69
LXII.	Distribución del curso <i>software</i> avanzado.....	78
LXIII.	Evaluación del curso de <i>software</i> avanzado.....	79
LXIV.	Detalle de sesiones del curso de <i>software</i> avanzado .....	79
LXV.	Sesión 1: Microsoft <i>Solutions Framework</i> .....	80
LXVI.	Sesión 2: modelo de equipo .....	81
LXVII.	Sesión 3: modelo de proceso .....	82
LXVIII.	Sesión 4: disciplina de gestión de riesgo .....	83
LXIX.	Sesión 5: análisis de riesgo .....	84
LXX.	Sesión 6: gestión de proyectos.....	84
LXXI.	Sesión 7: estimación.....	85
LXXII.	Sesión 8: puntos de función.....	85
LXXIII.	Sesión 9: planificación .....	86
LXXIV.	Sesión 10: modelo de costo total de propiedad .....	86
LXXV.	Sesión 11: retorno de inversión .....	87
LXXVI.	Sesión 12: RDI para inspección de <i>software</i> .....	87

LXXVII. Sesión 13: presupuestación.....	88
LXXVIII.Sesión 14: modelo diseño de la solución.....	88
LXXIX. Sesión 15: desarrollo y estabilización .....	89
LXXX. Sesión 16: TSP .....	89
LXXXI. Sesión 17: PSP .....	90
LXXXII. Sesión 18: administración de personal con PSP .....	90
LXXXIII.Sesión 19: tutorial PSP y TSP .....	91
LXXXIV.Sesión 20: sesión extra .....	91
LXXXV. Sesión 21: evaluación final.....	91
LXXXVI.Tutorial 1 .....	92
LXXXVII.Tutorial 2 .....	92
LXXXVIII.Tutorial 3 .....	93
LXXXIX.Tutorial 4 .....	94
XC. Tutorial 5 .....	94
XCI. Evaluaciones.....	95



## **GLOSARIO**

<b>Instructor Guideline</b>	En inglés Guía del Instructor, se define como una guía de laboratorio estructurada que define los objetivos, sesiones de clase, tareas y ponderación del mismo.
<b>EPS</b>	Ejercicio Profesional Supervisado.
<b>Módulo</b>	Unidad de medida por medio de la cual se agrupan todas las actividades relacionadas para poder impartir un curso.
<b>Sesión</b>	Se define como un conjunto de temas que se agrupan con el fin de ser impartidos por un instructor en una unidad determinada de tiempo.



## RESUMEN

Las tecnologías de la información y de las comunicaciones se encuentran en constante cambio, y como parte de este cambio surge la necesidad de realizar una actualización de los contenidos impartidos dentro de la carrera de Ingeniería en Ciencias y Sistemas, sin descuidar la calidad de la enseñanza impartida por esta institución.

Como parte de este proceso de actualización es necesario definir una herramienta que describa y detalle la estructura que debe llevar cada curso y laboratorio impartido en la carrera de Ciencias y Sistemas. Para este Ejercicio Profesional Supervisado (EPS), se definió dicha estructura por medio de una documentación detallada de los contenidos de los cursos y laboratorios de Análisis y Diseño de Sistemas 1, Análisis y Diseño de Sistemas 2 y *Software Avanzado*.

Para poder definir y estandarizar los laboratorios impartidos en la Escuela de Ingeniería en Ciencias y Sistemas, se tomó como base el concepto de guías de instructor utilizadas en el Centro Tecnológico de la India, siendo una guía de instructor un documento utilizado para administrar los contenidos de un laboratorio. El personal del Centro Tecnológico aportó sus valiosos conocimientos y participó en la supervisión de la creación de dichos documentos, basado en las experiencias en capacitación obtenidas por la empresa transnacional *TATA Consultancy Services*.

Se recibió un curso intensivo para conocer la metodología de elaboración de guías de instructor (documentos de planificación de laboratorios) durante cuatro meses, en el período de enero a abril del 2008. Con base a este conocimiento se elaboraron las guías de instructor para los cursos de Análisis y Diseño de Sistemas 1, Análisis y Diseño de Sistemas 2 y *Software Avanzado*.

Posteriormente a la creación de estos documentos se desarrollaron las guías para las clases de Análisis y Diseño de Sistemas 1, Análisis y Diseño de Sistemas 2 y *Software Avanzado*, por medio de un libro de texto que contiene las unidades más importantes de cada curso.

# OBJETIVOS

## General

Crear una herramienta para definir el esquema de estructuración de tres cursos impartidos por la Escuela de Ingeniería en Ciencias y Sistemas aplicando la metodología de administración de contenidos de laboratorios por medio de guías de instructor y proporcionar el material de apoyo de los cursos mediante un libro de texto.

## Específicos:

1. Elaborar una guía de instructor, para estructurar el contenido y las sesiones de un laboratorio, definiendo los criterios de evaluación y promoción de los laboratorios de los cursos:
  - Análisis y Diseño de Sistemas 1
  - Análisis y Diseño de Sistemas 2
  - *Software Avanzado*
2. Desarrollar el contenido de cada sesión de las guías de instructor en formato de presentaciones para facilitar su impartición y distribución.

3. Desarrollar documentación de apoyo para los cursos de Ingeniería en Ciencias y Sistemas desarrollando un texto de referencia que contenga las unidades más importantes de cada curso. Este material se desarrolló para los siguientes cursos:
  - Análisis y Diseño de Sistemas 1
  - Análisis y Diseño de Sistemas 2
  - *Software Avanzado*
  
4. Documentar una serie de artículos relacionados en cada unidad del curso, para el material de apoyo de clase.

## INTRODUCCIÓN

La Facultad de Ingeniería de la Universidad de San Carlos de Guatemala se ha caracterizado por formar profesionales de alta calidad que se desenvuelven de muy buena manera en el ámbito laboral. Como parte de este proceso de formación, es necesario garantizar que la calidad y el nivel de los profesionales permanezcan a lo largo del tiempo. Es por ello que surge la necesidad de realizar una actualización continua de los contenidos impartidos por la Escuela de Ingeniería en Ciencias y Sistemas, para que el nivel y prestigio de esta casa de estudios no decrezca.

Existe un segundo factor determinante en el éxito del profesional de sistemas, y es conocer las nuevas tendencias tecnológicas, por lo que es necesario establecer tanto un mecanismo de actualización de los contenidos impartidos al estudiante, como los elementos (documentos) que estructuren y faciliten este proceso de actualización

Adicionalmente, se observó que en la Escuela de Ingeniería en Ciencias y Sistemas era difícil coordinar el contenido impartido en un mismo curso en diferentes secciones, ya que no existía una guía definida de los temas a tratarse tanto en el curso como en el laboratorio, por lo que se decidió crear un sistema por medio del cual se pueda estandarizar el contenido de los cursos y laboratorios.

Por lo tanto, se desarrolló una solución a la problemática planteada anteriormente, definiendo un artefacto que permitiera actualizar los contenidos de los laboratorios de la carrera, permitir su extensibilidad a nuevas actualizaciones y hacer este proceso sencillo para su implementador. Para ello se trabajó en conjunto con la empresa *TATA CONSULTANCY SERVICES* y por medio del proyecto *India-Guatemala IT Education Centre of Excellence*, se brindó una capacitación a los estudiantes para poder realizar las guías de instructor. Mediante este curso transmitió al estudiante de EPS la metodología de elaboración de guías de instructor para administrar los contenidos de los laboratorios de la carrera de Ingeniería en Ciencias y Sistemas.

Se le indicó a la persona que trabajó en este proyecto que iba a tomar tres cursos de la carrera de Ingeniería en Ciencias y Sistemas y que iba a aplicar esta nueva metodología. Cada guía de instructor fue creada de acuerdo al curso impartido por el señor Mrutunjaya Panda, coordinador de Centro de Excelencia para la Enseñanza de las Tecnologías de la Información y la Comunicación. El contenido de los laboratorios fue sugerido y coordinado por los catedráticos titulares de cada uno de los cursos tratados en este documento.

En este documento Usted encontrará una breve descripción sobre la metodología de desarrollo de guías de instructor y en los capítulos posteriores encontrará la aplicación de esta metodología a los cursos de Análisis y Diseño de Sistemas 1, Análisis y Diseño de Sistemas 2 y *Software Avanzado*

Finalmente, encontrará la guía de contenidos de la documentación de apoyo generada para cada uno de los cursos mencionados anteriormente.

# **1. MARCO TEÓRICO**

## **1.1 Principios de pedagogía**

De la definición de los objetivos principales de este proyecto surge la necesidad de definir algunos conceptos básicos de pedagogía.

La pedagogía es la rama de la ciencia que se puede resumir en la frase “el arte de enseñar”. Esta ciencia busca tanto definir como aplicar las técnicas y prácticas que se realizan con el fin de facilitar el proceso de aprendizaje. Se busca optimizar el proceso de aprendizaje con el objetivo de beneficiar a ambos el educador y el objetivo a ser educado.

Ya que mediante esta ciencia estamos llegando al pensamiento humano, es imposible definir esta disciplina sin mencionar a la psicología. La psicología como ciencia individual busca estudiar los procesos mentales, mediante el estudio del comportamiento humano se pueden crear programas con el fin de optimizarlo.

De acuerdo a Chickerin y Gamson, los principios de la pedagogía son siete.

**a. Propiciar el contacto entre estudiantes y profesores.**

Aquí vemos el elemento fundamental de la psicología en la definición de la pedagogía. Es muy importante que el profesor logre crear un vínculo con sus estudiantes, la dinámica que se practica aquí varía dependiendo del número de estudiantes que se tiene. Si el estudiante se siente cómodo se logra crear un ambiente positivo en el cual es más fácil impartir el conocimiento.

**b. Fomentar la cooperación entre los estudiantes.**

Nunca se debe de olvidar que la interacción entre los estudiantes es muy importante en el proceso educativo. El profesor siempre debe incentivar el trabajo en equipo

**c. Propiciar el aprendizaje activo.**

El aprendizaje activo se refiere al grado de participación del estudiante. Este objetivo se puede derivar del primer y segundo objetivo. Si el estudiante siente un vínculo con el profesor y se ha creado un ambiente cooperativo entre los alumnos la participación del alumno viene de manera natural. La participación tiene como objetivo tanto ayudar al alumno como al profesor. De esta manera se logra una retroalimentación que se obtiene del alumno, tanto las opiniones como las vivencias pueden complementar el programa de estudios base en una clase.

**d. Proporcionar retroalimentación a tiempo.**

Es importante saber manejar la retroalimentación que se obtiene del estudiante y saber en qué casos es necesaria una respuesta inmediata. De esta manera se incentiva este tipo de comportamiento mejorando la sinergia del grupo completo.

**e. Enfatizar el uso apropiado del tiempo.**

La planeación del tiempo es crucial en el proceso de aprendizaje, los beneficios que obtenemos de una buena planeación van desde impartir el contenido deseado hasta aliviar la carga de trabajo en los alumnos. Tanto el alumno como el profesor se benefician de un programa bien definido en términos de tiempo. También es muy importante el manejo del tiempo en cada una de las sesiones de clase.

**f. Propiciar altas expectativas al estudiante.**

Al incentivar al estudiante a participar e intentar obtener un poco más de lo que se le pide normalmente a un alumno estamos creando metas más altas para él. Es importante hacerle saber al alumno que estas metas pueden alcanzarse y que los beneficios que se obtienen al lograr estas metas son muchos.

**g. Respetar los diferentes estilos de aprendizaje**

Este principio afirma que un programa de estudio, o un método de estudio no puede mantenerse estático. Es importante considerar la necesidad de cambiar e incluso aceptar otros estilos de aprendizaje. Mediante la observación activa podemos incorporar al programa básico de estudios ideas que podrían beneficiarlo.

## **1.2 Proceso de enseñanza – aprendizaje**

### **Educación**

Es un proceso social en el cual se busca que el alumno desarrolle sus capacidades mentales, técnicas o físicas para aumentar su habilidad en determinado campo o tema.

### **Enseñanza**

El sistema de enseñanza se basa en el modelo Estudiante-Profesor. El elemento llamado profesor tiene como objetivo impartir el conocimiento que posee sobre cierta área, la manera en que imparte dicho conocimiento demanda de él creatividad y habilidad. El segundo elemento en este modelo es llamado estudiante y es el sujeto al cual se le busca impartir conocimiento. Para que este sistema funcione se necesita del compromiso de ambos componentes.

### **Aprendizaje**

A su nivel más bajo se define como el proceso de adquirir conocimientos. La naturaleza de este conocimiento abarca tanto aspectos intelectuales como físicos y emocionales. Si el alumno logra aumentar dicho nivel de conocimiento podemos afirmar que se está logrando llegar al aprendizaje.

El tipo de aprendizaje que se busca en este proyecto es el aprendizaje intelectual, el objetivo que se busca es impartir conocimiento. Bajo este principio definimos lo que es el proceso de enseñanza-aprendizaje, en este proceso tenemos tres elementos principales. El primer elemento es el estudiante y el segundo es el alumno. El tercer elemento es lo que llamamos cuerpo de conocimiento, que se puede definir como el resultado del proceso de aprendizaje.

El cuerpo de conocimiento en sí es el conocimiento adquirido después de impartirse una clase, conocimiento que aumenta su capacidad mental o física.

Podemos ver que el proceso de enseñanza-aprendizaje es bastante complejo e involucra una gran cantidad de variables. Es por esa razón que resulta crucial tener una guía sobre la cual se pueda basar un plan para impartir conocimiento.

De los tres elementos principales descritos en el proceso enseñanza-aprendizaje existe tres interacciones directas que deben ser detalladas:

#### **Interacción profesor - conocimiento**

El elemento profesor debe tener claro cuáles son los recursos que tiene a su disposición para impartir conocimiento.

#### **Interacción alumno - conocimiento**

El alumno es un elemento crucial en este sistema y sin su colaboración no se puede completar el proceso de aprendizaje.

#### **Interacción profesor - alumno**

Es muy importante que el profesor pueda crear un vínculo con el estudiante para involucrarlo en el proceso y despertar en él un deseo de aprender.

### **Acción didáctica**

En el proceso de enseñanza-aprendizaje tenemos dos elementos que están íntimamente relacionados, en este proceso si definimos la acción por medio de la cual se imparte el conocimiento estamos describiendo a la acción didáctica. Para ejecutar la acción didáctica debemos conocer sus tres componentes principales:

1. **Planteamiento.** En esta etapa se definen los objetivos que se buscan lograr, a su vez definimos el conjunto de actividades que se deben realizar para orientarnos hacia dichos objetivos.

2. **Ejecución.** Basándonos en el plan descrito anteriormente y en los recursos que el profesor tiene a su disposición, se comienza el proceso de impartir conocimiento.

3. **Evaluación.** En esta etapa observamos si se han cumplido los objetivos propuestos. En base a esta observación podemos realizar cambios a nuestro planteamiento inicial.

### **1.3 Método de clases magistrales**

Existen diferentes maneras en que un profesor puede impartir una clase, podemos definir dos métodos principales. El primer método es lo que se llama una clase activa, en este tipo de método se incentiva al estudiante a participar en la clase. Esto se logra haciendo directamente preguntas a los estudiantes o incentivando la participación directa.

El segundo método se llama clase magistral, aquí se tiene un sistema diferente a la clase activa, ya que lo que se busca es que el profesor imparta una clase sin ser interrumpido por los estudiantes. El profesor comienza a dar la clase y los estudiantes pueden tomar nota de los elementos más importantes, pero su participación es pasiva ya que no se involucra directamente en la conversación.

Los beneficios principales de una clase magistral son la facilidad de administrar el tiempo y la capacidad de impartir mayor cantidad de material en una unidad de tiempo determinada. Es por esta razón que este método se utiliza con mayor frecuencia. Con este método es más fácil para el profesor preparar las sesiones a impartir.

Los elementos principales que debemos manejar durante una clase magistral son:

- **La voz**

Debemos tomar en cuenta el lugar en el cual se va a impartir la clase, si el lugar es muy grande es posible que tengamos necesidad de un micrófono. Por otro lado si el lugar que utilizamos es pequeño o la audiencia es de tamaño reducido tener un micrófono solamente molestaría al público.

- **El cuerpo**

El uso adecuado del cuerpo al realizar una exposición es muy importante. Es más fácil para el público seguir a una persona que está en constante movimiento, ya que evita que la exposición se vuelva aburrida. Por otro lado no debemos exagerar nuestros movimientos ya que el público pierde el interés en el contenido.

- **Las imágenes visuales**

El uso de diferentes ayudas visuales nos permite captar la atención del público con mayor facilidad y en muchas ocasiones facilita el entendimiento de algunos temas.

- **Invitados**

Traer como invitado a un experto en cierta materia puede ser beneficioso para una exposición.

- **El plan de exposición**

Es importante mostrarle al público al inicio cuál es el plan completo a seguir durante la exposición, ya que de esta manera puede saber de antemano que temas son de mayor interés para él.

- **El texto de la clase**

En algunas ocasiones puede ser de gran ayuda proveer al público con material escrito para que pueda seguir con mayor facilidad la exposición.

- **Resumen final**

Nos ayuda a repetir elementos importantes de la exposición.

## **1.4 Aprendizaje práctico por medio de laboratorios**

Existen ciertos temas que no pueden impartirse únicamente como una clase teórica, temas en los cuáles la parte práctica es tan importante o en algunos casos más importante. En este tipo de casos es una buena idea pensar en la utilización de un laboratorio, es una manera sencilla de involucrar directamente al estudiante y ayudarlo a comprender conceptos teóricos.

El lado negativo de este método es su alto costo en ciertos casos, si no se tienen los recursos adecuados y se tiene un grupo numeroso de estudiantes el laboratorio no es efectivo. Por esta razón se debe realizar un estudio detallado de quien será el grupo de estudiantes al cual el laboratorio está destinado y también se debe saber con qué tipo de equipo se puede contar.



## 2. DEFINICIÓN DE UNA GUÍA DEL INSTRUCTOR

Un *instructor guideline* es un documento que tiene como objetivo estructurar, estandarizar y detallar las diferentes actividades que se realizan al impartir una clase. Buscan también definir de manera clara el contenido que se impartirá y el tiempo en el cual se debe hacer.

El proceso de enseñanza requiere una planeación del tiempo bien definida, ya que es el elemento principal que se busca optimizar. Debemos saber qué tipo de actividades se deben realizar y con qué frecuencia se debe realizar cada una.

La planificación se detalla en un documento escrito, y dicha planeación se realiza con base a una serie de módulos bien definidos. La estructura de este documento y la manera en la cual se trabaja fue probada y utilizada por varios años por la empresa TATA.

Cada *instructor guideline* va dirigido directamente a la persona que está encargada de impartir la clase, ya sea un auxiliar o un catedrático. Cada una de las secciones de este documento provee una herramienta importante cuya finalidad es facilitar la labor de impartir clases.

La planificación del proceso de enseñanza debe poseer las siguientes características:

- **Flexibilidad.** Debemos tener en cuenta que los planes que se hacen pueden ser adaptados a los cambios que sean necesarios.

- **Realidad.** Se debe saber exactamente con que recursos se cuenta en el lugar que se imparte una clase, a partir de este conocimiento se puede crear un plan efectivo para lograr los objetivos planteados para cada módulo.

- **Precisión.** Cada guía debe definir cada elemento de manera clara y precisa. Si se da lugar a la ambigüedad se puede llegar a tener problemas de entendimiento y por consecuencia la guía se aleja de los objetivos planteados.

## **2.1 Objetivos de una guía de instructor**

Los objetivos principales de la guía del instructor son estructurar, ordenar, estandarizar y especificar el contenido que se utiliza para impartir una clase o un laboratorio.

Esta guía permite llevar el control con más facilidad de un curso cuando se tiene diferentes secciones u horarios. Evita la necesidad de verificar constantemente entre los diferentes catedráticos sobre el contenido y el ritmo impartido en el curso. Al tener un punto de referencia claro sobre el cual respaldarse es más fácil llevar el control del curso y evitar problemas de tiempo y de contenido debido a la planeación.

## **2.2 Elementos de una guía de instructor**

Una guía de instructor está compuesta por varios elementos, estos son los componentes principales:

- Información general del curso.
- Distribución de horas y actividades.
- Forma de evaluación.
- Detalle de sesiones.
- Distribución de sesiones.
- Detalle de tareas asignadas

- Detalle de tutoriales
- Detalle de exámenes.
- Ejemplo de exámenes.
- Bibliografía.
- Anexos.

## **2.3 Proceso de construcción de la guía del instructor**

### **Información general del curso**

Este elemento trata de describir la idea general e introductoria del curso, esta debe contener:

- Nombre del curso, según el pensum, incluye el código del curso.
- Prerrequisitos, los cursos previos que son requisitos obligatorios.
- Objetivo, la meta a alcanzar para el curso.
- Módulos, componentes del curso con diferente finalidad.

### **Distribución de horas y actividades.**

En esta sección especificamos cuantos módulos se trabajan por curso y se detalla para cada uno las diferentes actividades que se realizan y la cantidad en horas de tiempo que se tarda el instructor en impartirla.

Las actividades que se detallan son: asistencias, tareas, presentaciones, y evaluaciones.

La siguiente tabla muestra un ejemplo de la distribución de horas y actividades:

**Tabla I. Distribución de horas y actividades**

<b>MÓDULO</b>	<b>TEORÍA</b>	<b>PRÁCTICA</b>	<b>TOTAL</b>	<b>EXAMEN</b>	<b>TUTORIALES</b>	<b>PRESENTACIONES</b>
Módulo 1	30 horas	8 horas	38 horas	2 horas	4	2
<b>TOTAL</b>	<b>30 horas</b>	<b>8 horas</b>	<b>38 horas</b>	<b>2 horas</b>	<b>4</b>	<b>2</b>

#### **Forma de evaluación.**

Se le asigna un valor numérico a cada actividad que se realizará para llegar a un 100% de la nota total. Aquí se tiene una tabla de ejemplo:

**Tabla II. Forma de evaluación**

<b>ACTIVIDAD</b>	<b>NÚMERO</b>	<b>PONDERACIÓN</b>	<b>TOTAL</b>
PRÁCTICAS	2	10%	20%
TAREAS	2	10%	20%
PRESENTACIONES	2	20%	40%
EVALUACIÓN	1	20%	20%
		<b>TOTAL</b>	<b>100%</b>

**Detalle de sesiones.**

Se define el número de sesiones en las cuales se divide cada módulo del curso, se muestra también la duración, definiendo si existirá alguna tarea en cada sesión.

**Tabla III. Detalle de sesiones**

<b>NO.</b>	<b>NOMBRE DE SESIÓN</b>	<b>TIPO DE SESIÓN</b>		<b>TIPO DE ACTIVIDAD</b>
		TEÓRICA	PRÁCTICA	
1	Nombre sesión	X		
2	Nombre sesión		X	Presentación

### **Distribución de sesiones.**

Previo a la realización de una sesión para un módulo se deben tomar en cuenta una serie de factores diferentes. El primer factor es la naturaleza del contenido que se debe incluir, para considerar esto se debe saber que cursos ha llevado anteriormente el estudiante para establecer su base de conocimiento. Otro factor importante es el tiempo de cada sesión, cada sesión está programada para impartirse en dos horas, de manera que se debe estimar el grado de dificultad del contenido o si habrá necesidad de mostrar varios ejemplos para explicar bien el tema. Finalmente se especifica en cada sesión si se estará haciendo un examen y cuál debe ser su duración.

La distribución de sesiones se realiza dividiendo el módulo completo en una cantidad fija de sesiones, en el documento debemos identificar el nombre del curso, el módulo al que pertenece, el número de sesión, el objetivo, la distribución de temas y el objetivo del módulo.

Las sesiones pueden ser teóricas o prácticas, donde una sesión teórica constituye una clase magistral, y una sesión práctica un ejercicio que se realizará durante el período de clase.

Por cada sesión teórica definida en la guía del instructor, se debe desarrollar una presentación con el contenido de dicha sesión, para obtener tanto la definición de los puntos a tratar en la sesión, y el contenido de la sesión en sí.

### **Detalle de tareas asignadas**

En este módulo se dividen de manera similar a la distribución de sesiones las diferentes tareas que se deben realizar. Se detalla aquí: el módulo al que pertenece, el objetivo que se busca, la forma de evaluación, el material a entregar y finalmente en qué momento se debe hacer.

### **Detalle de tutoriales**

En esta sección se describen los objetivos y sistema de evaluación de los tutoriales (conocidos como prácticas presenciales) que se realizarán en el laboratorio.

### **Detalle de exámenes.**

Se define aquí la manera en que se debe hacer un examen para el módulo. Se detalla aquí: el módulo al que pertenece, cuando se debe realizar, el contenido a evaluar, la duración y el modo de evaluación.

### **Ejemplo de exámenes.**

Para tener una mejor idea de los objetivos que se buscan al realizar una prueba se muestra en esta sección un ejemplo completo de un examen. Por medio de un ejemplo es más fácil apreciar de qué manera se debe crear la evaluación. Representa solamente un ejemplo de una evaluación, cada auxiliar de laboratorio deberá elaborar sus propios exámenes durante cada semestre.

### **Bibliografía.**

Es el conjunto de material didáctico que se utiliza como base para creación del contenido incluido en cada sesión. También se detalla para tener un texto de referencia en caso de necesitar ampliar la información dada.

### **Anexos**

En esta sección se incluyen elementos extras que complementan la guía de instructor. Pueden incluirse tutoriales de ejemplo, el programa del laboratorio y otros elementos que se consideren necesarios para mejorar la comprensión de la guía.

## **2.4 Forma de uso de la guía de instructor**

Cada guía de instructor tiene como objetivo ayudar a la persona que imparte las sesiones de cada módulo. Esta ayuda abarca varios aspectos entre ellos: administración del tiempo al detallar cada sesión, el contenido a abarcar en cada sesión, el número de ejercicios y evaluaciones que se deben realizar, la manera en que se deben calificar y los objetivos que se buscan para cada módulo.

### **3. GUÍA DEL INSTRUCTOR DE ANÁLISIS Y DISEÑO DE SISTEMAS 1**

A continuación se presenta el desarrollo de la guía del curso de Análisis y Diseño de Sistemas 1, de acuerdo al formato sugerido en el curso de “Estructuración de laboratorios”, impartido por el señor. Mrutunjaya Panda.

#### **3.1 Información general**

NOMBRE DE CURSO: Análisis y Diseño de Sistemas 1 (283)

PRE- REQUISITOS: Sistemas de Bases de Datos 1 (774)

POST – REQUISITOS: Análisis y Diseño de Sistemas 2 (785)

Sistemas Organizacionales y Gerenciales 1 (786)

Sistemas Aplicados 1 (788)

OBJETIVO: Comprender los conceptos básicos referentes a la ingeniería de software y al proceso de desarrollo de software, teniendo entre estos: el ciclo de desarrollo de software, las metodologías de desarrollo más

comúnmente utilizadas, herramientas para toma y análisis de requerimientos y la fase de pruebas del software.

MÓDULOS: Ingeniería de Software I

NOTA DE PROMOCIÓN: 61 puntos

### 3.2 Distribución de horas y actividades

**Tabla IV. Distribución del curso Análisis y Diseño de Sistemas 1**

<b>MODULO</b>	<b>TEORÍA</b>	<b>PRÁCTICA</b>	<b>TOTAL</b>	<b>EXAMEN</b>	<b>TUTORIALES</b>	<b>PRESENTACIONES</b>
INGENIERÍA DE SOFTWARE I	30 horas	8 horas	38 horas	2 horas	4	2
<b>TOTAL</b>	<b>30 horas</b>	<b>8 horas</b>	<b>38 horas</b>	<b>2 horas</b>	<b>4</b>	<b>2</b>

### 3.3 Evaluación

Ponderación de tareas, prácticas y proyectos.

**Tabla V. Evaluación del curso de Análisis y Diseño de Sistemas 1**

<b>ACTIVIDAD</b>	<b>NUMERO</b>	<b>PONDERACIÓN</b>	<b>TOTAL</b>
PRÁCTICAS	2	10.00 %	20.00 %
TAREAS	2	10.00 %	20.00 %
PRESENTACIONES Y TRABAJO ESCRITO	2	20.00 %	40.00 %
EVALUACIÓN	1	20.00 %	20.00 %
		<b>TOTAL</b>	<b>100.00 %</b>

### 3.4 Detalle de sesiones

**Tabla VI. Detalle de sesiones del curso de Análisis y Diseño de Sistemas 1**

NO	NOMBRE DE SESIÓN	TIPO DE SESIÓN		TIPO DE ACTIVIDAD
		TEÓRICA	PRÁCTICA	
1	CONCEPTOS DE SISTEMAS DE INFORMACIÓN	X		
2	PROCESO DE DESARROLLO DE SOFTWARE	X		
3	MODELOS DE DESARROLLO DE SOFTWARE	X		
4	METODOLOGÍAS RÍGIDAS	X		
5	METODOLOGÍAS ÁGILES	X		
6	EXTREME PROGRAMMING	X		
7	DSDM	X		
8	OPEN SOURCE	X		
9	PRESENTACIONES DE ESTUDIANTES: "METODOLOGÍAS DE DESARROLLO"		X	PRESENTACIONES
10	CASOS DE USO	X		
11	EJERCICIOS DE CASOS DE USO		X	PRÁCTICA NO. 1
12	REQUERIMIENTOS NO FUNCIONALES	X		
13	USABILIDAD	X		
14	HISTORIAS DE USUARIO	X		
15	EJERCICIOS DE HISTORIAS DE USUARIO		X	PRÁCTICA NO. 2
16	DISCIPLINA DE PRUEBAS DE RUP	X		
17	MODELO CMMI	X		
18	ITIL	X		
19	PRESENTACIONES DE ESTUDIANTES: "PROCESO DE PRUEBAS DEL SOFTWARE"		X	PRESENTACIONES
20	SESIÓN EXTRA			
21	EVALUACIÓN FINAL		X	EVALUACIÓN

### 3.5 Distribución de sesiones

**Tabla VII. Sesión 1: conceptos de sistemas de información**

SESIÓN 1:	CONCEPTOS DE SISTEMAS DE INFORMACIÓN
<b>OBJETIVO:</b>	<b>Entender los conceptos generales y características y principales tipos de sistemas de información.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Sistemas               <ol style="list-style-type: none"> <li>1.1. Definición</li> </ol> </li> <li>2. Componentes de sistemas               <ol style="list-style-type: none"> <li>2.1. Información y datos</li> <li>2.2. Recursos</li> <li>2.3. Datos intermedios</li> <li>2.4. Procedimientos</li> <li>2.5. Procesos</li> </ol> </li> <li>3. Características de los sistemas               <ol style="list-style-type: none"> <li>3.1.1. Objetivo</li> <li>3.1.2. Estándares</li> <li>3.1.3. Ambiente</li> <li>3.1.4. Retroalimentación</li> <li>3.1.5. Límites e interfaces</li> </ol> </li> <li>4. Sistemas de información               <ol style="list-style-type: none"> <li>4.1. Definición</li> <li>4.2. Tipos de sistemas de información                   <ol style="list-style-type: none"> <li>4.2.1. TPS</li> <li>4.2.2. MIS</li> <li>4.2.3. DSS</li> </ol> </li> </ol> </li> <li>5. Tipos de sistemas en tiempo real               <ol style="list-style-type: none"> <li>5.1. Sistemas críticos</li> <li>5.2. Sistemas acrílicos</li> <li>5.3. Sistemas con parada segura</li> <li>5.4. Sistemas con degradación aceptable</li> </ol> </li> <li>6. Sistemas de emisión crítica               <ol style="list-style-type: none"> <li>6.1.1. Sistemas seguros frente al exterior</li> <li>6.1.2. Sistemas críticos de misión</li> <li>6.1.3. Sistemas críticos de negocios</li> </ol> </li> </ol>

**Tabla VIII. Sesión 2: proceso de desarrollo de software**

<b>SESIÓN 2:</b>	<b>PROCESO DE DESARROLLO DE SOFTWARE</b>
<b>OBJETIVO:</b>	<b>Analizar y entender las fases del ciclo de desarrollo de software así como las principales ventajas y desventajas de los diferentes modelos de desarrollo de software.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Ciclo de desarrollo de software               <ol style="list-style-type: none"> <li>1.1. Definición</li> </ol> </li> <li>2. Fases del ciclo de desarrollo               <ol style="list-style-type: none"> <li>2.1. Investigación preliminar</li> <li>2.2. Análisis</li> <li>2.3. Diseño</li> <li>2.4. Desarrollo</li> <li>2.5. Pruebas</li> <li>2.6. Implementación y mantenimiento</li> </ol> </li> <li>3. Modelos de desarrollo de software               <ol style="list-style-type: none"> <li>3.1. Cascada                   <ol style="list-style-type: none"> <li>3.1.1. Definición</li> </ol> </li> <li>3.2. Modelo incremental                   <ol style="list-style-type: none"> <li>3.2.1. Definición</li> </ol> </li> </ol> </li> </ol>

**Tabla IX. Sesión 3: metodologías de desarrollo de software**

<b>SESIÓN 3:</b>	<b>METODOLOGÍAS DE DESARROLLO DE SOFTWARE</b>
<b>OBJETIVO:</b>	<b>Comprender las principales ventajas y desventajas de los diferentes modelos de desarrollo de software.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Modelos de desarrollo de software               <ol style="list-style-type: none"> <li>1.1. Modelo de prototipos                   <ol style="list-style-type: none"> <li>1.1.1. Definición</li> <li>1.1.2. Prototipo</li> </ol> </li> <li>1.2. Modelo espiral                   <ol style="list-style-type: none"> <li>1.2.1. Definición</li> <li>1.2.2. Fases                       <ol style="list-style-type: none"> <li>1.2.2.1. Comunicación con el cliente</li> <li>1.2.2.2. Planificación</li> <li>1.2.2.3. Análisis de riesgo</li> <li>1.2.2.4. Ingeniería</li> <li>1.2.2.5. Construcción y acción</li> <li>1.2.2.6. Evaluación del cliente</li> </ol> </li> </ol> </li> <li>1.3. Comparación entre diferentes modelos</li> </ol> </li> </ol>

**Tabla X. Sesión 4: metodologías rígidas**

SESIÓN 4:	METODOLOGÍAS RÍGIDAS
<b>OBJETIVO:</b>	<b>Comprender que son las metodologías de desarrollo de software, así como las bases de la metodología de desarrollo RUP.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Metodologías de desarrollo de software               <ol style="list-style-type: none"> <li>1.1. Definición</li> </ol> </li> <li>2. Metodologías rígidas               <ol style="list-style-type: none"> <li>2.1. Definición</li> </ol> </li> <li>3. <i>Rational Unified Process</i> <ol style="list-style-type: none"> <li>3.1. Descripción</li> <li>3.2. Fases                   <ol style="list-style-type: none"> <li>3.2.1. <i>Inception</i></li> <li>3.2.2. <i>Elaboration</i></li> <li>3.2.3. <i>Construction</i></li> <li>3.2.4. <i>Transition</i></li> </ol> </li> <li>3.3. Disciplinas                   <ol style="list-style-type: none"> <li>3.3.1. Modelado del negocio</li> <li>3.3.2. Requerimientos</li> <li>3.3.3. Análisis y diseño</li> <li>3.3.4. Implementación</li> <li>3.3.5. Pruebas</li> <li>3.3.6. Despliegue</li> <li>3.3.7. Administración de la configuración y cambios</li> <li>3.3.8. Administración de proyecto</li> <li>3.3.9. Ambiente</li> </ol> </li> <li>3.4. Ventajas y desventajas de la metodología</li> </ol> </li> </ol>

**Tabla XI. Sesión 5: metodologías ágiles**

SESIÓN 5:	METODOLOGÍAS ÁGILES
<b>OBJETIVO:</b>	<b>Comprender que es la ingeniería de software y conocer los fundamentos de las metodologías ágiles de desarrollo.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Ingeniería de Software               <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. Áreas de la administración de la ingeniería de <i>software</i></li> <li>1.3. Disciplinas relacionadas con la ingeniería de <i>software</i></li> </ol> </li> <li>2. Metodologías Ágiles de Desarrollo               <ol style="list-style-type: none"> <li>2.1. Definición</li> <li>2.2. Principios de las Metodologías Ágiles</li> <li>2.3. Valores del manifiesto ágil</li> <li>2.4. <i>Manifiesto for agile software development</i></li> </ol> </li> </ol>

**Tabla XII. Sesión 6: *Extreme Programming***

SESIÓN 6:	<b>EXTREME PROGRAMMING</b>
<b>OBJETIVO:</b>	<b>Comprender los aspectos fundamentales de la metodología de desarrollo <i>Extreme Programming</i>.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. <i>Extreme Programming</i> <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. Fases                             <ol style="list-style-type: none"> <li>1.2.1. <i>Exploration</i></li> <li>1.2.2. <i>Planning</i></li> <li>1.2.3. <i>Iterations to Release</i></li> <li>1.2.4. <i>Productionizing</i></li> <li>1.2.5. <i>Maintenance</i></li> <li>1.2.6. <i>Death</i></li> </ol> </li> <li>1.3. Valores de <i>Extreme Programming</i> <ol style="list-style-type: none"> <li>1.3.1. Comunicación</li> <li>1.3.2. Simplicidad</li> <li>1.3.3. Retroalimentación</li> <li>1.3.4. Coraje</li> <li>1.3.5. Respeto</li> </ol> </li> <li>1.4. Principios                             <ol style="list-style-type: none"> <li>1.4.1. Rápida retroalimentación</li> <li>1.4.2. Asumir simplicidad</li> <li>1.4.3. Cambio incremental</li> <li>1.4.4. Abrazar el cambio</li> <li>1.4.5. Trabajo de calidad</li> </ol> </li> <li>1.5. Mejores prácticas de <i>Extreme Programming</i> <ol style="list-style-type: none"> <li>1.5.1. Escala fina de retroalimentación                                     <ol style="list-style-type: none"> <li>1.5.1.1. Programar en parejas</li> <li>1.5.1.2. Planificar el juego</li> <li>1.5.1.3. Desarrollo dirigido por las pruebas</li> <li>1.5.1.4. Equipo completo</li> </ol> </li> <li>1.5.2. Proceso continuo                                     <ol style="list-style-type: none"> <li>1.5.2.1. Integración continua</li> <li>1.5.2.2. Mejoras de diseño</li> <li>1.5.2.3. Pequeños entregables</li> </ol> </li> <li>1.5.3. Compartiendo el conocimiento                                     <ol style="list-style-type: none"> <li>1.5.3.1. Estándares de codificación</li> <li>1.5.3.2. Código de propiedad colectiva</li> <li>1.5.3.3. Diseño simple</li> <li>1.5.3.4. Metáfora del sistema</li> </ol> </li> <li>1.5.4. Bienestar del programador                                     <ol style="list-style-type: none"> <li>1.5.4.1. Paso sostenible</li> </ol> </li> </ol> </li> <li>1.6. Ventajas y desventajas de la metodología</li> </ol> </li> </ol>

**Tabla XIII. Sesión 7: *Dynamics System Development Method***

<b>SESIÓN 7:</b>	<b><i>DYNAMICS SYSTEM DEVELOPMENT METHOD</i></b>
<b>OBJETIVO:</b>	<b>Comprender las diferencias entre las distintas metodologías de desarrollo.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. DSDM             <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. RAD</li> <li>1.3. Fases                 <ol style="list-style-type: none"> <li>1.3.1. <i>Feasibility Study</i></li> <li>1.3.2. <i>Business Study</i></li> <li>1.3.3. <i>Functional Model Iteration</i></li> <li>1.3.4. <i>Design and Build Iteration</i></li> <li>1.3.5. <i>Implementation</i></li> </ol> </li> <li>1.4. Principios                 <ol style="list-style-type: none"> <li>1.4.1. Involucrar al usuario de forma activa es imperativo</li> <li>1.4.2. Los equipos deben de ser autorizados para la toma de decisiones</li> <li>1.4.3. El enfoque es en la entrega frecuente de productos</li> <li>1.4.4. La salud para los propósitos del negocio es el criterio esencial para la liberación de productos</li> <li>1.4.5. El desarrollo iterativo e incremental es necesario para converger en una solución de negocios precisa</li> <li>1.4.6. Todos los cambios durante el desarrollo son reversibles</li> <li>1.4.7. Los requerimientos son definidos a alto nivel</li> <li>1.4.8. Las pruebas son integradas a través del ciclo de vida</li> <li>1.4.9. Una aproximación cooperativa y colaborativa entre <i>stakeholders</i> es esencial</li> </ol> </li> <li>1.5. Técnicas de DSDM                 <ol style="list-style-type: none"> <li>1.5.1. Cajas de tiempo</li> <li>1.5.2. MoSCoW</li> <li>1.5.3. Prototipado</li> <li>1.5.4. Pruebas</li> <li>1.5.5. <i>Workshop</i></li> <li>1.5.6. Modelado</li> <li>1.5.7. Administración de la configuración</li> </ol> </li> <li>1.6. Asunciones</li> <li>1.7. Ventajas y desventajas de la metodología</li> </ol> </li> </ol>

**Tabla XIV. Sesión 8: Open Source Software**

<b>SESIÓN 8:</b>	<b>OPEN SOURCE SOFTWARE</b>
<b>OBJETIVO:</b>	<b>Comprender los aspectos fundamentales de la metodología de desarrollo <i>Open Source</i>.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Metodología de desarrollo <i>Open Source</i> <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. Fases                             <ol style="list-style-type: none"> <li>1.2.1. <i>Problem discovery</i></li> <li>1.2.2. <i>Finding volunteers</i></li> <li>1.2.3. <i>Solution identification</i></li> <li>1.2.4. <i>Code development and testing</i></li> <li>1.2.5. <i>Code change review</i></li> <li>1.2.6. <i>Code commit and documentation</i></li> <li>1.2.7. <i>Release management</i></li> </ol> </li> <li>1.3. Ventajas y desventajas de la metodología</li> <li>1.4. Escenarios donde se recomienda</li> </ol> </li> <li>2. Resumen de metodologías</li> </ol>

**Tabla XV. Sesión 9: presentaciones: “metodologías de desarrollo”**

<b>SESIÓN 9:</b>	<b>PRESENTACIONES: “METODOLOGÍAS DE DESARROLLO”</b>
<b>OBJETIVO:</b>	<b>Presentar temas de investigación.</b>
<b>CONSIDERACIONES:</b>	<ol style="list-style-type: none"> <li>1. Las presentaciones pueden ser realizadas en grupos, de acuerdo al número de estudiantes del curso</li> <li>2. Cada presentación debe de durar al menos 10 minutos</li> <li>3. Cada grupo debe de presentar un trabajo escrito sobre el tema expuesto</li> <li>4. El contenido de las presentaciones debe de ser asignado por el auxiliar del curso y el mismo debe de ser relacionado con el ciclo de desarrollo de software o metodologías de desarrollo de software, para complementar los temas de laboratorio cubiertos.</li> <li>5. Como sugerencias se presentan los siguientes temas (pueden cubrirse estos u otros que el auxiliar cree de importancia para enriquecer el conocimiento de los estudiantes)             <ul style="list-style-type: none"> <li>• Scrum</li> <li>• FDD</li> <li>• Crystal</li> <li>• Highsmith ASD</li> <li>• SQUARE</li> <li>• <i>Agile Unified Process</i></li> <li>• Métodos de desarrollo de <i>software</i> en Guatemala</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>• Técnicas usadas para documentar requerimientos en Guatemala</li> </ul>
--	---

**Tabla XVI. Sesión 10: casos de uso**

<b>SESIÓN 10:</b>	<b>CASOS DE USO</b>
<b>OBJETIVO:</b>	<b>Comprender qué son los casos de uso, sus componentes y sea capaz de construir casos de uso a partir de un conjunto de requerimientos del usuario.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Requerimientos funcionales</li> <li>2. Casos de uso <ol style="list-style-type: none"> <li>2.1. Objetivos de los casos de uso</li> <li>2.2. Modelo de caso de uso <ol style="list-style-type: none"> <li>2.2.1. Elementos <ol style="list-style-type: none"> <li>2.2.1.1. Actores</li> <li>2.2.1.2. Casos de uso</li> </ol> </li> <li>2.2.2. Representación UML</li> </ol> </li> <li>2.3. Flujos en casos de uso <ol style="list-style-type: none"> <li>2.3.1. Flujo básico</li> <li>2.3.2. Flujo alterno</li> </ol> </li> <li>2.4. Precondición y postcondición</li> <li>2.5. Relaciones entre casos de uso <ol style="list-style-type: none"> <li>2.5.1. <i>Include</i> <ol style="list-style-type: none"> <li>2.5.1.1. Definición</li> <li>2.5.1.2. Ejemplo</li> </ol> </li> <li>2.5.2. <i>Extend</i> <ol style="list-style-type: none"> <li>2.5.2.1. Definición</li> <li>2.5.2.2. Ejemplo</li> </ol> </li> <li>2.5.3. Generalización <ol style="list-style-type: none"> <li>2.5.3.1. Definición</li> <li>2.5.3.2. Ejemplo</li> </ol> </li> </ol> </li> </ol> </li> </ol>

**Tabla XVII. Sesión 11: ejercicios de casos de uso**

<b>SESIÓN 11:</b>	<b>EJERCICIOS DE CASOS DE USO</b>
<b>OBJETIVO:</b>	<b>Realizar la definición de Casos de Uso para un caso de estudio en particular.</b>
<b>CONTENIDO:</b>	Durante esta sesión, los estudiantes deberán de realizar el diseño de casos de uso para un caso de estudio en particular. El mismo se realizará en grupos.

**Tabla XVIII. Sesión 12: requerimientos no funcionales**

<b>SESIÓN 12:</b>	<b>REQUERIMIENTOS NO FUNCIONALES</b>
<b>OBJETIVO:</b>	<b>Mostrar la importancia de los requerimientos no funcionales y su impacto en el producto final.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Requerimientos no funcional</li> <li>2. FURPS             <ol style="list-style-type: none"> <li>2.1. Funcionalidad                 <ol style="list-style-type: none"> <li>2.1.1. Definición</li> <li>2.1.2. Impacto en el producto final</li> </ol> </li> <li>2.2. Usabilidad                 <ol style="list-style-type: none"> <li>2.2.1. Definición</li> <li>2.2.2. Impacto en el producto final</li> </ol> </li> <li>2.3. Fiabilidad                 <ol style="list-style-type: none"> <li>2.3.1. Definición</li> <li>2.3.2. Impacto en el producto final</li> </ol> </li> <li>2.4. Rendimiento                 <ol style="list-style-type: none"> <li>2.4.1. Definición</li> <li>2.4.2. Impacto en el producto final</li> </ol> </li> <li>2.5. Soportabilidad                 <ol style="list-style-type: none"> <li>2.5.1. Definición</li> <li>2.5.2. Impacto en el producto final</li> </ol> </li> </ol> </li> <li>3. Otros requerimientos no funcionales             <ol style="list-style-type: none"> <li>3.1. Seguridad                 <ol style="list-style-type: none"> <li>3.1.1. Definición</li> <li>3.1.2. Impacto en el producto final</li> </ol> </li> <li>3.2. Portabilidad                 <ol style="list-style-type: none"> <li>3.2.1. Definición</li> <li>3.2.2. Impacto en el producto final</li> </ol> </li> <li>3.3. Modificación                 <ol style="list-style-type: none"> <li>3.3.1. Definición</li> <li>3.3.2. Impacto en el producto final</li> </ol> </li> </ol> </li> </ol>

**Tabla XIX. Sesión 13: usabilidad**

<b>SESIÓN 13:</b>	<b>USABILIDAD</b>
<b>OBJETIVO:</b>	<b>Mostrar al estudiante la importancia de la usabilidad en los proyectos de software y los aspectos más importantes que la misma abarca.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Principios de la usabilidad</li> <li>2. Usabilidad Web</li> <li>3. Elementos principales de la Usabilidad Web               <ol style="list-style-type: none"> <li>3.1. Accesibilidad</li> <li>3.2. Psicología del color</li> <li>3.3. Facilidad de uso (Intuitivo)</li> <li>3.4. Estructura (esquema de la página)</li> <li>3.5. Navegación</li> <li>3.6. Gráficos</li> <li>3.7. Tipografía</li> </ol> </li> <li>4. Evaluar la usabilidad</li> </ol>

**Tabla XX. Sesión 14: historias de usuario**

<b>SESIÓN 14:</b>	<b>HISTORIAS DE USUARIO</b>
<b>OBJETIVO:</b>	<b>Comprender los elementos principales necesarios para construir una historia de usuario.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Historias de usuario           <ol style="list-style-type: none"> <li>1.1. Definición de una historia de usuario</li> <li>1.2. Características de las historias de usuario</li> <li>1.3. ¿Qué no es una historia de usuario?</li> <li>1.4. Proceso de planificación de historias de usuario</li> </ol> </li> <li>2. Partes de una historia de usuario</li> <li>3. Spyke</li> </ol>

**Tabla XXI. Sesión 15: ejercicios de historias de usuario**

<b>SESIÓN 15:</b>	<b>EJERCICIOS DE HISTORIAS DE USUARIO</b>
<b>OBJETIVO:</b>	<b>Poner en práctica los conceptos principales de desarrollo de historias de usuario mediante un caso de estudio.</b>
<b>CONTENIDO:</b>	Durante esta sesión, los estudiantes deberán de diseñar las historias de usuario para un caso de estudio en particular. El mismo se realizará en grupos.

**Tabla XXII. Sesión 16: disciplina de pruebas de RUP**

<b>SESIÓN 16:</b>	<b>DISCIPLINA DE PRUEBAS DE RUP</b>
<b>OBJETIVO:</b>	<b>Comprender los aspectos más importantes de la disciplina de pruebas de RUP.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Disciplina de pruebas de RUP             <ol style="list-style-type: none"> <li>1.1. Propósito</li> <li>1.2. Filosofía de la disciplina de pruebas de RUP                 <ol style="list-style-type: none"> <li>1.2.1. Desarrollo iterativo</li> <li>1.2.2. Aproximación holística</li> <li>1.2.3. Automatización</li> </ol> </li> <li>1.3. Medición de pruebas</li> <li>1.4. Cobertura de las pruebas                 <ol style="list-style-type: none"> <li>1.4.1. Cobertura de las pruebas basadas en requerimientos</li> <li>1.4.2. Cobertura de las pruebas Basadas en código</li> </ol> </li> <li>1.5. Calidad percibida por medición</li> <li>1.6. Reporte de defectos                 <ol style="list-style-type: none"> <li>1.6.1. Reportes de densidad</li> <li>1.6.2. Reportes de añejamiento</li> <li>1.6.3. Reportes de tendencia</li> </ol> </li> <li>1.7. Mediciones de performance                 <ol style="list-style-type: none"> <li>1.7.1. Monitoreo dinámico</li> <li>1.7.2. Tiempos de respuesta</li> </ol> </li> <li>1.8. Roles y actividades</li> <li>1.9. <i>Workflow</i></li> <li>1.10. Principales artefactos                 <ol style="list-style-type: none"> <li>1.10.1. Plan de pruebas</li> <li>1.10.2. Resumen de evaluación de pruebas</li> <li>1.10.3. Casos de prueba</li> <li>1.10.4. Configuración del ambiente de pruebas</li> </ol> </li> </ol> </li> </ol>

**Tabla XXIII. Sesión 17: modelo CMMI**

<b>SESIÓN 17:</b>	<b>MODELO CMMI</b>
<b>OBJETIVO:</b>	<b>Comprender el modelo de CMMI y su relación con la calidad del software.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. CMMI               <ol style="list-style-type: none"> <li>1.1. Introducción</li> <li>1.2. Modelos CMMI                   <ol style="list-style-type: none"> <li>1.2.1. CMMI-DEV</li> <li>1.2.2. CMMI-SVC</li> <li>1.2.3. CMMI-ACQ</li> </ol> </li> <li>1.3. Niveles de CMM                   <ol style="list-style-type: none"> <li>1.3.1. Inicial</li> <li>1.3.2. Gestionado</li> <li>1.3.3. Definido</li> <li>1.3.4. Cuantitativamente gestionado</li> <li>1.3.5. Optimizado</li> </ol> </li> <li>1.4. Componentes del modelo</li> <li>1.5. Áreas de proceso                   <ol style="list-style-type: none"> <li>1.5.1. Gestión de proceso</li> <li>1.5.2. Gestión de proyecto</li> <li>1.5.3. Ingeniería</li> <li>1.5.4. Soporte</li> </ol> </li> <li>1.6. CMMI-DEV                   <ol style="list-style-type: none"> <li>1.6.1. Alcance</li> <li>1.6.2. Área de proceso</li> </ol> </li> <li>1.7. Modelo de referencia de la medición de la calidad del producto</li> <li>1.8. Elementos del modelo de calidad</li> <li>1.9. ISO 25000</li> </ol> </li> </ol>

**Tabla XXIV. Sesión 18: ITIL**

<b>SESIÓN 18:</b>	<b>ITIL</b>
<b>OBJETIVO:</b>	<b>Conocer una introducción sobre el marco de trabajo ITIL y su relación con la calidad del <i>software</i>.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. ITIL             <ol style="list-style-type: none"> <li>1.1. Introducción</li> <li>1.2. Etapas del ciclo de vida de ITIL                 <ol style="list-style-type: none"> <li>1.2.1. Estrategia del servicio</li> <li>1.2.2. Diseño del servicio</li> <li>1.2.3. Transición del servicio</li> <li>1.2.4. Operación del servicio</li> <li>1.2.5. Mejora continua del servicio</li> </ol> </li> <li>1.3. Los ocho libros de ITIL</li> <li>1.4. Un marco de trabajo para control                 <ol style="list-style-type: none"> <li>1.4.1. Definición</li> <li>1.4.2. Medición</li> <li>1.4.3. Análisis</li> <li>1.4.4. Mejora</li> <li>1.4.5. Control</li> </ol> </li> </ol> </li> <li>2. ISO 20000</li> </ol>

**Tabla XXV. Sesión 19: presentaciones proceso de pruebas de *software***

<b>SESIÓN 19:</b>	<b>PRESENTACIONES: “PROCESO DE PRUEBAS DE SOFTWARE”</b>
<b>OBJETIVO:</b>	<b>Presentar temas de investigación.</b>
<b>CONSIDERACIONES:</b>	<ol style="list-style-type: none"> <li>1. Las presentaciones pueden ser realizadas en grupos, de acuerdo al número de estudiantes del curso</li> <li>2. Cada presentación debe de durar al menos 10 minutos</li> <li>3. Cada grupo debe de presentar un trabajo escrito sobre el tema expuesto</li> <li>4. El contenido de las presentaciones debe de ser relacionado con el proceso de pruebas y aseguramiento de la calidad del <i>software</i>. El contenido que se expondrá debe de ser de aporte para la vida profesional del estudiante. El mismo deberá de ser asignado por el auxiliar o dejado a discreción del estudiante.</li> </ol>

**Tabla XXVI. Sesión 20: sesión extra**

<b>SESIÓN 20:</b>	<b>SESIÓN EXTRA</b>
<b>OBJETIVO:</b>	Esta sesión se deja para reponer alguna sesión que no se haya podido impartir por algún motivo en particular.

**Tabla XXVII. Sesión 21: evaluación final**

<b>SESIÓN 21:</b>	<b>EVALUACIÓN FINAL</b>
<b>OBJETIVO:</b>	Durante esta sesión, se realizará la evaluación final del laboratorio.

### 3.6 Detalle de tareas asignadas

**Tabla XXVIII. Asignación de Laboratorio 1**

<b>ASIGNACIÓN:</b>	<b>Asignación de Laboratorio 1</b>
<b>OBJETIVO:</b>	<b>Que el estudiante ponga en práctica los conceptos aprendidos sobre casos de uso por medio de un ejemplo práctico</b>
Iniciado en la sesión de clase número:	10
Realizado en la sesión de clase número:	11
Demo a ser conducido:	No aplica
Criterio de evaluación:	Identificación de actores .....30% Identificación de casos de uso .....30% <i>Outline</i> de casos de uso.....30% Relación <i>include</i> y <i>extend</i> .....10%
Documentación a obtener o resultado esperado:	Diagrama de casos de uso y <i>outline</i> de casos de uso

**Tabla XXIX. Asignación de Laboratorio 2**

<b>ASIGNACIÓN:</b>	<b>Asignación de Laboratorio 2</b>
<b>OBJETIVO:</b>	<b>Evaluar el grado de comprensión del tema historias de usuario, mediante el desarrollo de las historias de usuario para un caso de estudio en particular.</b>
Iniciado en la sesión de clase número:	14
Realizado en la sesión de clase número:	15
Demo a ser conducido:	No aplica
Criterio de evaluación:	Identificación de las historias de usuario.....35% Definición de las historias de usuario .....35% Integración con Spyke .....30%
Documentación a obtener o resultado esperado:	Historias de usuario y su integración con Spyke

### 3.7 Detalle de tutoriales

Tabla XXX. Tutorial 1

<b>TUTORIAL:</b>	<b>Tutorial 1</b>
<b>OBJETIVO:</b>	Elaborar un tutorial sobre características de las metodologías de desarrollo.
Iniciado en la sesión de clase número:	2
Completado en las sesión de clase número:	8
Demo a ser conducido:	No aplica
Criterio de evaluación:	Comparación entre distintas metodologías..... 50% Aplicación de la teoría a un caso de estudio..... 50%
Documentación a obtener o resultado esperado:	Tutorial resuelto

**Tabla XXXI. Tutorial 2**

<b>TUTORIAL:</b>	<b>Tutorial 2</b>
<b>OBJETIVO:</b>	Elaborar un documento de estándares de usabilidad para la web (aplicable a su proyecto de clase).
Iniciado en la sesión de clase número:	13
Completado en las sesión de clase número:	13
Demo a ser conducido:	No aplica
Criterio de evaluación:	Implementación de la teoría de usabilidad..... 50% Aplicado al proyecto de clase .....50%
Documentación a obtener o resultado esperado:	Documento de estándares de usabilidad para la web

### 3.8 Detalle de evaluaciones

**Tabla XXXII. Evaluaciones**

<b>NÚMERO TOTAL DE EVALUACIONES:</b>	1
<b>EVALUACIÓN NÚMERO:</b>	1
Evaluación a ser tomada después de la sesión número:	20
Temas de la evaluación:	Ingeniería de <i>Software</i> I (módulo completo)
Duración de la evaluación:	2 hrs.

Distribución de la evaluación:	Parte teórica	- 40 %
	Parte práctica	- 60%

### 3.9 Evaluación de ejemplo

#### Parte teórica (40 Pts.):

1. ¿Cuáles son los tipos de sistemas de información?
2. ¿Qué son los sistemas de emisión crítica?
3. ¿Cuáles son las fases del ciclo de desarrollo?
4. ¿Qué son las metodologías ágiles de desarrollo?
5. ¿Qué es la usabilidad?
6. ¿Qué es el *performance*?
7. ¿Qué es CMMI?
8. ¿Cuáles son los niveles de CMM?
9. ¿Qué es ITIL?
10. ¿Cuáles son los cinco elementos de ITIL?

#### Parte práctica (60 Pts.):

##### Desarrollo de temas (50 Pts.)

1. Explique de forma breve en qué consiste el modelo de cascada y realice un diagrama del mismo
2. Explique de forma breve en qué consiste la metodología de desarrollo RUP
3. Explique de forma breve los aspectos importantes de XP
4. Explique de forma breve los aspectos fundamentales y principales artefactos de la disciplina de pruebas de RUP
5. Realice un cuadro comparativo entre las disciplinas DSDM, XP y RUP

### **Caso práctico (10 Pts.)**

Realice el diagrama de casos de uso para el siguiente caso de estudio:

Se desea construir un sistema de información para llevar el control pagos de una empresa a sus proveedores y empleados. El sistema deberá de enviar al banco con el que la empresa trabaje la definición de todas las operaciones bancarias para realizar los pagos de salarios y proveedores mediante lotes de pagos semanales o quincenales (cada lote de pago constituye un conjunto de pagos).

El proceso a seguirse será el siguiente:

El contador de la empresa crea un lote de pagos (ya sea planilla o pago a proveedores), en el cual se definen a que personas se les pagará y el monto a acreditar en sus cuentas.

Una vez el lote de pagos es registrado, el mismo es almacenado en el sistema (pero no es enviado al banco todavía).

Una vez creado el lote de pagos, el mismo puede ser modificado por el contador por si fuera necesario agregar o eliminar pagos dentro del lote. El asistente financiero también puede modificar el lote, pero no crearlo, ya que su función no es crear, sino solamente editar el mismo. Además, el auditor de la empresa puede modificar el lote.

El asistente de gerencia puede consultar el estado de los lotes del sistema (si ya fueron enviados al banco o no) y puede ver el contenido de los lotes de pago, pero no puede modificar la información.

El sistema enviará el lote de pagos al banco de forma automática. Para que un lote pueda ser enviado, el mismo deberá de ser aprobado por las siguientes personas: gerente financiero y asistente financiero donde el gerente financiero no puede modificar el lote, solamente puede ver el

estado y contenido del lote, mientras que el asistente financiero no tiene los privilegios para modificar los datos del lote.

Una vez el lote ha sido autorizado por ambas personas, el mismo será enviado al banco y el estatus del lote de pago cambiará a enviado, ya no siendo posible modificarlo por ninguna persona.

### 3.10 Bibliografía recomendada

1. Presman, Roger S. **Ingeniería de Software, un enfoque práctico**. 5ª ed. Editorial McGraw Hill.
2. Bass, Len, Paul, Clements y Rick, Kazman. **Software Architecture in Practice**. 2ª ed. Editorial Addison Wesley.
3. Kroll, Per y Kruchten, Philippe. **The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP**. Editorial Addison Wesley, 2003. 464 pp.
4. Abran, Alain y Moore, James. **Guide to the software engineering body of knowledge**. IEEE Computer Society, 2004. 200 pp.
5. Sharma, Srinarayan y otros. **A framework for creating hybrid-open source software communities**. Artículo, Estados Unidos de Norteamérica universidad de Oakland, departamento de las ciencias, escuela de administración de negocios.
6. Leavitt, Michael y Shneiderman, Ben. **Research-Based Web Design & Usability Guideline. Introduction to the Architecture of the CMMI Framework**. Nota técnica, Software Engineering Institute, Universidad de Carnegie Mellon.

7. **Capability Maturity Model Integration (CMMISM), Version 1.1.**  
Nota técnica, universidad Carnegie Mellon, Software Engineering Institute.
8. Proxima Technology. **Establishing a Quality Framework for ITIL, an Overview of Six Sigma and its Role in Itil Service Delivery.** Nota Técnica. Versión 1.0
9. Catlidge, Alison y otros. **An Introductory Overview of ITIL V3.** itSMF Ltd.
10. [www.dsdm.org](http://www.dsdm.org) (agosto de 2008)
11. <http://agilemanifesto.org/> (agosto de 2008)
12. [www.extremeprogramming.org](http://www.extremeprogramming.org) (agosto de 2008)
13. <http://www.sei.cmu.edu/cmmi/> (agosto de 2008)
14. <http://www.itil.org/> (agosto de 2008)

## **3.11 Anexos**

### **A. Recomendaciones**

1. Es recomendable entregar al principio del semestre la calendarización de actividades (entrega de tareas y presentaciones) a los alumnos del laboratorio para evitar contratiempos posteriores.
2. Es recomendable realizar la asignación de los temas de investigación al inicio del semestre, para evitar contratiempos para la realización de los mismos.
3. Es recomendable explicar a los alumnos al inicio de semestre que actividades se realizarán de forma individual y que otras se realizarán de forma grupal, y cuando este sea el caso, definir desde un inicio el número máximo de integrantes por grupo.
4. Es necesario tener un medio de comunicación con los alumnos, adicional al correo electrónico, por lo que se recomienda registrarse en la Universidad Virtual, o crear un grupo para mantener la comunicación con los alumnos y publicar documentos.

## **B. Tutoriales**

Los tutoriales son tareas, investigaciones o prácticas que los estudiantes deben de realizar con el fin de complementar el contenido del laboratorio, o bien poner en práctica los conocimientos adquiridos en el mismo.

### **Tutorial 1**

#### **Objetivo**

El tutorial 1 tiene como objetivo evaluar el grado de comprensión de los estudiantes sobre las metodologías de desarrollo.

#### **Temas a evaluar**

- Sistemas de información
- Modelos del ciclo de desarrollo de software
- Manifiesto de desarrollo ágil
- Metodologías de desarrollo
  - RUP
  - XP
  - DSDM
  - *Open Source*

### Tutorial de ejemplo

1. Describa en forma breve y de un ejemplo de cada uno de los tipos de sistemas de información
2. Explique en qué consisten cada una de las fases del ciclo de desarrollo de software
3. Complete el siguiente cuadro comparativo entre los distintos modelos de desarrollo

**Tabla XXXIII. Cuadro comparativo “modelos del ciclo de desarrollo”  
(tutorial de ejemplo)**

	<b>Modelo de cascada</b>	<b>Modelo espiral</b>
Elementos principales		
Ventajas		
Desventajas		
Escenarios donde puede aplicarse		
Diferencias entre ambos modelos		

4. Explique en qué consisten las fase de concepción y elaboración de la metodología de desarrollo RUP
5. Complete el siguiente cuadro comparativo

**Tabla XXXIV. Cuadro comparativo “metodologías de desarrollo” (tutorial de ejemplo)**

	<b>Metodología de desarrollo RUP</b>
En qué consiste la metodología	
Cuales son su fases	
Elementos que la distinguen de las demás metodologías	
Ventajas frente a XP y DSDM	
Desventajas de la metodología	
Escenarios donde puede aplicarse	
	<b>Metodología de desarrollo XP</b>
En qué consiste la metodología	
Cuales son su fases	
Elementos que la distinguen de las demás metodologías	
Ventajas frente a RUP y DSDM	
Desventajas de la metodología	
Escenarios donde puede aplicarse	
	<b>Metodología de desarrollo DSDM</b>
En qué consiste la metodología	
Cuales son su fases	
Elementos que la distinguen de las demás metodologías	
Ventajas frente a RUP y XP	
Desventajas de la metodología	
Escenarios donde puede aplicarse	

6. Caso de estudio:

Se le ha propuesto como encargado de administrar el desarrollo de un sistema para auditoría y control de mercaderías procedentes de la china, las cuales arribarán por el litoral Atlántico a nuestro país.

El sistema debe de ser desarrollado en un período no mayor a dos meses, el cual al cabo de dicho tiempo debe de ser completamente funcional y no debe de contar con ningún tipo de errores.

Usted tiene a su cargo 5 programadores para dicha tarea, y un analista/arquitecto para diseñar la solución del producto.

Se conoce de antemano que los clientes encargados de definir las reglas del negocio tienen por costumbre pedir sesiones periódicas para revisiones del avance del producto en el proceso de desarrollo.

Debido a la naturaleza del producto y el constante cambio en las políticas aduaneras, los requerimientos del negocio pueden sufrir leves cambios durante el desarrollo del producto.

Proponga una metodología de desarrollo que usted considera eficiente para realizar el desarrollo del sistema en cuestión y explique por qué escogió esa metodología.

## **Tutorial 2**

### **Objetivo**

Elaborar un documento de estándares de usabilidad para la web (aplicable al proyecto de clase).

### **Metodología de trabajo**

El estudiante deberá de investigar los estándares de usabilidad para la web, documentarlos y definir los estándares que utilizará en su proyecto, justificándolos con la investigación realizada.

### **Entregable**

Un documento que identifique que aspectos de la usabilidad se usarán en su proyecto. El mismo debe de incluir un marco teórico y la explicación de la parte de su proyecto donde lo aplicarán.

### **Observaciones**

Los tutoriales fueron pensados para dejarse como tarea de casa, por lo que puede realizarse de forma individual, o en grupos según el auxiliar de laboratorio crea conveniente.

El objetivo del tutorial es que el estudiante refuerce sus conocimientos sobre los temas del laboratorio, por lo que el estudiante se encuentra en absoluta libertad de solventar dudas sobre los tutoriales con el auxiliar del curso.



## 4. GUÍA DEL INSTRUCTOR DE ANÁLISIS Y DISEÑO DE SISTEMAS 2

A continuación se presenta el desarrollo de la guía del curso de Análisis y Diseño de Sistemas 2, de acuerdo al formato sugerido en el curso de “Estructuración de laboratorios”, impartido por el señor. Mrutunjaya Panda.

### 4.1 Información general

NOMBRE DE CURSO: Análisis y Diseño de Sistemas 2 (785)

PRE- REQUISITOS: Análisis y Diseño de Sistemas 1 (283)

POST – REQUISITOS: Software Avanzado (780)

Sistemas Aplicados 2 (789)

Auditoría de Proyectos de Software (735)

OBJETIVO: Comprender los aspectos generales referentes a la arquitectura de software, incluyendo algunos métodos de integración de sistemas, que sea capaz de administrar un sistema para versionar productos de *software* y conozca el proceso de

aseguramiento de la calidad del *software*.

MÓDULOS: Ingeniería de Software II

NOTA DE PROMOCIÓN: 61 puntos

## 4.2 Distribución de horas y actividades

**Tabla XXXV. Distribución del curso Análisis y Diseño de Sistemas 2**

<b>MODULO</b>	<b>TEORÍA</b>	<b>PRÁCTICA</b>	<b>TOTAL</b>	<b>EXAMEN</b>	<b>TUTORIALES</b>	<b>PRESENTACIONES</b>
INGENIERÍA DE SOFTWARE II	34 horas	4 horas	38 horas	2 horas	3	2
<b>TOTAL</b>	<b>34 horas</b>	<b>4 horas</b>	<b>38 horas</b>	<b>2 horas</b>	<b>3</b>	<b>2</b>

## 4.3 Evaluación

Ponderación de tareas, prácticas y proyectos.

**Tabla XXXVI. Evaluación del curso de Análisis y Diseño de Sistemas 2**

ACTIVIDAD	NUMERO	PONDERACIÓN	TOTAL
ASISTENCIAS O EXAMENES CORTOS	15	1.00 %	15.00 %
TUTORIALES	3	10.00 %	30.00 %
PRESENTACIONES CON TRABAJO ESCRITO	2	15.00 %	30.00 %
EVALUACIÓN	1	25.00 %	25.00 %
		<b>TOTAL</b>	<b>100.00 %</b>

#### 4.4 Detalle de sesiones

**Tabla XXXVII. Detalle de sesiones del curso de Análisis y Diseño de Sistemas 2**

NO	NOMBRE DE SESIÓN	TIPO DE SESIÓN		TIPO DE ACTIVIDAD
		TEÓRICA	PRÁCTICA	
1	ADMINISTRACIÓN DE LA CONFIGURACIÓN	X		
2	VERSIONAMIENTO DE SISTEMAS DE INFORMACIÓN	X		
3	PLAN DE LA ADMINISTRACIÓN DE LA CONFIGURACIÓN	X		
4	REQUERIMIENTOS NO FUNCIONALES	X		
5	PRESENTACIONES DE ESTUDIANTES: "HERRAMIENTAS PARA ADMINISTRACIÓN DE LA CONFIGURACIÓN"		X	PRESENTACIONES
6	CONFIGURACIÓN DE UNA HERRAMIENTA PARA ADMINISTRACIÓN DE LA CONFIGURACIÓN	X		
7	SPRING	X		
8	MAPEO OBJETO RELACIONAL	X		
9	IMPLEMENTACION DE ORM EN DIVERAS TECNOLOGÍAS	X		
10	MODELO VISTA CONTROLADOR	X		

11	ARQUITECTURA DE SOFTWARE	X		
12	SERVIDOR DE APLICACIONES	X		
13	COMPONENTES EN SERVIDORES DE APLICACIONES	X		
14	PRUEBAS DE SOFTWARE	X		
15	PRESENTACIONES DE ESTUDIANTES: "HERRAMIENTAS PARA PRUEBAS"		X	PRESENTACIONES
16	INTEGRACIÓN DE SISTEMAS	X		
17	MESSAGE BROKER	X		
18	HERRAMIENTA MESSAGE BROKER	X		
19	SESIÓN EXTRA			
20	EVALUACIÓN FINAL		X	EVALUACIÓN

#### 4.5 Distribución de sesiones

**Tabla XXXVIII. Sesión 1: administración de la configuración**

<b>SESIÓN 1:</b>	<b>ADMINISTRACIÓN DE LA CONFIGURACIÓN</b>
<b>OBJETIVO:</b>	<b>Lograr la comprensión de los aspectos que se deben de tomar en cuenta para la administración de la configuración y su implementación</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Administración de la configuración <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. Aspectos operacionales para la administración de la configuración <ol style="list-style-type: none"> <li>1.2.1. Identificación</li> <li>1.2.2. Control</li> <li>1.2.3. Contabilidad de estados</li> <li>1.2.4. Auditoría y revisión</li> </ol> </li> <li>1.3. Terminología</li> </ol> </li> <li>2. Sistema de control de versiones <ol style="list-style-type: none"> <li>2.1. Definición</li> <li>2.2. Áreas Funcionales de un sistema para control de versiones <ol style="list-style-type: none"> <li>2.2.1. Componentes</li> <li>2.2.2. Estructura</li> <li>2.2.3. Construcción</li> <li>2.2.4. Auditoría</li> <li>2.2.5. Contabilidad</li> <li>2.2.6. Control</li> <li>2.2.7. Proceso</li> </ol> </li> </ol> </li> </ol>

	2.2.8. Team
--	-------------

**Tabla XXXIX. Sesión 2: versionamiento de sistemas de información**

<b>SESIÓN 2:</b>	<b>VERSIONAMIENTO DE SISTEMAS DE INFORMACIÓN</b>
<b>OBJETIVO:</b>	<b>Lograr la comprensión de los aspectos que se deben de tomar en cuenta para la administración de la configuración y su implementación</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Herramientas para control de versiones               <ol style="list-style-type: none"> <li>1.1. Subversion</li> <li>1.2. Maven</li> <li>1.3. SourceSafe</li> </ol> </li> <li>2. Modelos de administración de fuentes               <ol style="list-style-type: none"> <li>2.1. Bloqueo de documentos</li> <li>2.2. Unión de versiones</li> </ol> </li> <li>3. Tipos de repositorios               <ol style="list-style-type: none"> <li>3.1. Centralizado</li> <li>3.2. Distribuido</li> </ol> </li> <li>4. Plan de la administración de la configuración               <ol style="list-style-type: none"> <li>4.1. Pasos para elaborar un plan para administración de la configuración</li> <li>4.2. Ejemplos de numeraciones para versiones</li> </ol> </li> </ol>

**Tabla XL. Sesión 3: plan de la administración de la configuración**

<b>SESIÓN 3:</b>	<b>PLAN DE LA ADMINISTRACIÓN DE LA CONFIGURACIÓN</b>
<b>OBJETIVO:</b>	<b>Desarrollar el plan de configuración del proyecto de clase</b>
<b>CONTENIDO:</b>	Durante esta sesión, el auxiliar deberá de explicar la plantilla <i>Configuration Management Plan</i> de RUP

**Tabla XLI. Sesión 4: requerimientos no funcionales**

<b>SESIÓN 4:</b>	<b>REQUERIMIENTOS NO FUNCIONALES</b>
<b>OBJETIVO:</b>	<b>Comprender la importancia de evaluar los requerimientos no funcionales al momento de diseñar un sistema y su impacto en el mismo al momento de ser implementado.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Requerimientos no funcionales</li> <li>2. ISO 9126</li> <li>3. Algunos atributos de calidad               <ol style="list-style-type: none"> <li>3.1. Funcionalidad                   <ol style="list-style-type: none"> <li>3.1.1. Impacto en los sistemas de información</li> </ol> </li> <li>3.2. Usabilidad                   <ol style="list-style-type: none"> <li>3.2.1. Usabilidad Web</li> </ol> </li> <li>3.3. Fiabilidad                   <ol style="list-style-type: none"> <li>3.3.1. Impacto en los sistemas críticos</li> </ol> </li> <li>3.4. Performance                   <ol style="list-style-type: none"> <li>3.4.1. Impacto en los sistemas de información</li> </ol> </li> <li>3.5. Soporte                   <ol style="list-style-type: none"> <li>3.5.1. Impacto en los sistemas durante su ciclo de vida</li> </ol> </li> <li>3.6. Disponibilidad                   <ol style="list-style-type: none"> <li>3.6.1. Impacto en los sistemas críticos (7x24)</li> </ol> </li> <li>3.7. Escalabilidad                   <ol style="list-style-type: none"> <li>3.7.1. Impacto en los sistemas de alto crecimiento</li> </ol> </li> <li>3.8. Seguridad                   <ol style="list-style-type: none"> <li>3.8.1. Impacto en los sistemas Web</li> </ol> </li> </ol> </li> </ol>

**Tabla XLII. Sesión 5: presentaciones de estudiantes: “herramientas para administración de la configuración”**

<b>SESIÓN 5:</b>	<b>PRESENTACIONES DE ESTUDIANTES: “HERRAMIENTAS PARA ADMINISTRACIÓN DE LA CONFIGURACIÓN”</b>
<b>OBJETIVO:</b>	<b>Conocer algunas de las herramientas que existen para administración de la configuración.</b>
<b>CONTENIDO:</b>	<p>Se realizarán presentaciones en grupos sobre diversas herramientas para administración de la configuración, indicando las ventajas y desventajas de las mismas, con el objetivo que los estudiantes formen un criterio para elegir que herramienta utilizar en el proyecto de clase.</p> <p>Dentro de la exposición, cada grupo deberá mostrar de forma práctica, como se implementa la herramienta para administración de la configuración registrando diferentes versiones de un mismo archivo en el repositorio central.</p> <p>Cada grupo deberá de realizar un trabajo escrito que será entregado al</p>

	<p>auxiliar de cátedra y a los alumnos del curso.</p> <p>Los grupos deberán de ser formados acorde al número de estudiantes del curso, y pueden ser los mismos grupos que se formaron para el proyecto de clase.</p>
--	--

**Tabla XLIII. Sesión 6: configuración de una herramienta para administración de la configuración**

<b>SESIÓN 6:</b>	<b>CONFIGURACIÓN DE UNA HERRAMIENTA PARA ADMINISTRACIÓN DE LA CONFIGURACIÓN</b>
<b>OBJETIVO:</b>	<b>Aprender a configurar un sistema para administración de la configuración.</b>
<b>CONTENIDO:</b>	Durante esta sesión, el auxiliar deberá mostrar al estudiante como configurar una herramienta para la administración de la configuración. Se sugiere SubVersion para Java y Microsoft Source Safe para .NET

**Tabla XLIV. Sesión 7: *SPRING***

<b>SESIÓN 7:</b>	<b><i>SPRING</i></b>
<b>OBJETIVO:</b>	<b>Conocer los principales elementos que conforman el marco de trabajo <i>Spring</i>.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Spring <ol style="list-style-type: none"> <li>1.1. Descripción</li> <li>1.2. Principio de inversión de control (IoC)</li> <li>1.3. Principio de dependencia de inyección (DI)</li> </ol> </li> <li>2. <i>Spring framework</i> <ol style="list-style-type: none"> <li>2.1. <i>Core</i></li> <li>2.2. <i>Context</i></li> <li>2.3. DAO</li> <li>2.4. ORM</li> <li>2.5. AOP</li> <li>2.6. Web</li> <li>2.7. MVC</li> </ol> </li> <li>3. Escenarios comunes <ol style="list-style-type: none"> <li>3.1.1. Web</li> <li>3.1.2. Capas intermedias (<i>Middlewares</i>)</li> <li>3.1.3. Métodos remotos</li> </ol> </li> <li>4. JDKs que lo soportan</li> <li>5. <i>Spring 2.0</i> <ol style="list-style-type: none"> <li>5.1. Contenedor IoC</li> </ol> </li> </ol>

	5.2. AOP 5.3. Capa intermedia 5.4. JPA 5.5. Capa Web
--	---

**Tabla XLV. Sesión 8: mapeo objeto relacional**

<b>SESIÓN 8:</b>	<b>MAPEO OBJETO RELACIONAL</b>
<b>OBJETIVO:</b>	<b>Introducir al estudiante en los conceptos de mapeo de objetos relacionales a clases mediante ORM y su implementación en una tecnología específica.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. <i>Object-Relational Mapping</i> <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. Fundamentos</li> <li>1.3. Aspectos importantes</li> <li>1.4. Mapeo de la herencia</li> <li>1.5. Escenarios donde se recomienda</li> </ol> </li> <li>2. ORM en Java <ol style="list-style-type: none"> <li>2.1. Hibernate</li> <li>2.2. Otros productos</li> </ol> </li> <li>3. ORM en .NET <ol style="list-style-type: none"> <li>3.1. NHibernate</li> <li>3.2. Otros productos</li> </ol> </li> <li>4. Ventaja y desventajas</li> </ol>

**Tabla XLVI. Sesión 9: implementación de ORM en diversas tecnologías**

<b>SESIÓN 9:</b>	<b>IMPLEMENTACIÓN DE ORM EN DIVERSAS TECNOLOGÍAS</b>
<b>OBJETIVO:</b>	<b>Comprender de forma práctica como se implementa Hibernate</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Hibernate               <ol style="list-style-type: none"> <li>1.1. Descripción</li> <li>1.2. Componentes necesarios para implementarlo (pre-requisitos)</li> <li>1.3. Archivo de configuración</li> <li>1.4. POJO</li> <li>1.5. Forma de uso                   <ol style="list-style-type: none"> <li>1.5.1. Mapeo de objetos a tablas</li> <li>1.5.2. Actualización</li> <li>1.5.3. Transacciones</li> </ol> </li> <li>1.6. Dialectos de SQL soportados</li> </ol> </li> <li>2. NHibernate               <ol style="list-style-type: none"> <li>2.1. Descripción</li> <li>2.2. Componentes necesarios para implementarlo (pre-requisitos)</li> <li>2.3. Archivo de configuración</li> <li>2.4. POCO</li> <li>2.5. Forma de uso                   <ol style="list-style-type: none"> <li>2.5.1. Mapeo de objetos a tablas</li> <li>2.5.2. Actualización</li> <li>2.5.3. Transacciones</li> </ol> </li> <li>2.6. Dialectos de SQL soportados</li> </ol> </li> </ol>
<b>PARTE PRÁCTICA:</b>	Durante esta sesión, el auxiliar mostrará a los estudiantes mediante ejemplos sencillos como implementar Hibernate (Java) y NHibernate(.Net)

**Tabla XLVII. Sesión 10: modelo vista controlador**

<b>SESIÓN 10:</b>	<b>MODELO VISTA CONTROLADOR</b>
<b>OBJETIVO:</b>	<b>Comprender los aspectos fundamentales del modelo vista controlador</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Modelo vista controlador             <ol style="list-style-type: none"> <li>1.1. Descripción del patrón</li> <li>1.2. Principio <i>open-close</i></li> <li>1.3. Componentes                 <ol style="list-style-type: none"> <li>1.3.1. El modelo</li> <li>1.3.2. Vista</li> <li>1.3.3. Controlador</li> </ol> </li> <li>1.4. Representación UML</li> </ol> </li> <li>2. Ejemplo</li> <li>3. <i>Frameworks MVC</i> <ol style="list-style-type: none"> <li>3.1. Java</li> <li>3.2. .Net</li> </ol> </li> </ol>

**Tabla XLVIII. Sesión 11: arquitectura de software**

<b>SESIÓN 11:</b>	<b>ARQUITECTURA DE SOFTWARE</b>
<b>OBJETIVO:</b>	<b>Comprender los aspectos fundamentales de la arquitectura de software correspondientes al diseño de componentes, paquetes y clases.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Arquitectura del software             <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. Modelo 4+1 vistas</li> <li>1.3. Vista lógica                 <ol style="list-style-type: none"> <li>1.3.1. Componentes</li> <li>1.3.2. Paquetes</li> <li>1.3.3. Clases</li> </ol> </li> <li>1.4. Arquitectura de 3 capas                 <ol style="list-style-type: none"> <li>1.4.1. Capa de presentación</li> <li>1.4.2. Capa de lógica del negocio</li> <li>1.4.3. Capa de acceso a datos</li> </ol> </li> </ol> </li> <li>2. Ejemplo</li> </ol>
<b>PARTE PRÁCTICA:</b>	Al finalizar la presentación, el auxiliar deberá elaborar la arquitectura software para un caso de estudio, mostrando a los estudiantes la forma como se diseña un sistema

**Tabla XLIX. Sesión 12: servidor de aplicaciones**

<b>SESIÓN 12:</b>	<b>SERVIDOR DE APLICACIONES</b>
<b>OBJETIVO:</b>	<b>Conocer los pasos fundamentales para configura un servidor de aplicaciones y un servidor web, tanto para java como para .net</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Servidor de aplicaciones               <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. Servidores de aplicaciones para Java</li> <li>1.3. Servidores de aplicaciones para .Net</li> </ol> </li> <li>2. Servidor Web               <ol style="list-style-type: none"> <li>2.1. Definición</li> <li>2.2. Servidores Web para Java y .net</li> </ol> </li> <li>3. Configuraciones               <ol style="list-style-type: none"> <li>3.1. Configuraciones para Java                   <ol style="list-style-type: none"> <li>3.1.1. Servidor de aplicaciones Jboss                       <ol style="list-style-type: none"> <li>3.1.1.1. Configuración del <i>Classpath</i></li> <li>3.1.1.2. JRE y JDK</li> </ol> </li> <li>3.1.2. Servidor Web                       <ol style="list-style-type: none"> <li>3.1.2.1. Apache Tomcat</li> <li>3.1.2.2. Extensión Axis</li> </ol> </li> </ol> </li> <li>3.2. Configuraciones para .Net                   <ol style="list-style-type: none"> <li>3.2.1. Servicio de componentes Com+</li> <li>3.2.2. Servidor Web IIS                       <ol style="list-style-type: none"> <li>3.2.2.1. Configurar IIS para utilizar el <i>framework</i> 2.0</li> </ol> </li> </ol> </li> </ol> </li> </ol>

**Tabla L. Sesión 13: componentes en servidores de aplicaciones**

<b>SESIÓN 13:</b>	<b>COMPONENTES EN SERVIDORES DE APLICACIONES</b>
<b>OBJETIVO:</b>	<b>Conocer la forma como se encapsula la lógica del negocio en módulos y la publicación de la misma en el servidor de aplicaciones.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Creación de módulos               <ol style="list-style-type: none"> <li>1.1. J2EE                   <ol style="list-style-type: none"> <li>1.1.1. <i>Enterprise Java Beans</i> <ol style="list-style-type: none"> <li>1.1.1.1. Encapsulamiento del código en componentes Jar</li> </ol> </li> <li>1.1.2. Servicios Web y JSP                       <ol style="list-style-type: none"> <li>1.1.2.1. Encapsulamiento del código en componentes War</li> </ol> </li> <li>1.1.3. Creación de módulos para J2EE                       <ol style="list-style-type: none"> <li>1.1.3.1. Encapsulamiento del código en componentes Ear</li> </ol> </li> </ol> </li> <li>1.2. .Net                   <ol style="list-style-type: none"> <li>1.2.1. Creación de componentes de librería</li> <li>1.2.2. Importar referencias                       <ol style="list-style-type: none"> <li>1.2.2.1. Diferencias entre referencias DLL y COM+</li> </ol> </li> </ol> </li> </ol> </li> <li>2. Registrar componentes en servidores de aplicaciones               <ol style="list-style-type: none"> <li>2.1. J2EE                   <ol style="list-style-type: none"> <li>2.1.1. Jboss</li> </ol> </li> </ol> </li> </ol>

	2.2. .Net 2.2.1.Com+
<b>SUGERENCIA:</b>	Si el auxiliar de laboratorio lo desea, puede realizar una demostración de la forma como se crean y publican los EJBs, módulos War y Ear, así como los componentes COM en los servidores de aplicaciones

**Tabla LI. Sesión 14: pruebas de software**

<b>SESIÓN 14:</b>	<b>PRUEBAS DE SOFTWARE</b>
<b>OBJETIVO:</b>	<b>Conocer los principales tipos de pruebas para sistemas de información y la forma como se elaboran las pruebas de los sistemas de información.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Proceso de pruebas de software <ol style="list-style-type: none"> <li>1.1. Pruebas unitarias</li> <li>1.2. Pruebas de integración</li> <li>1.3. Pruebas del sistema</li> </ol> </li> <li>2. Casos de prueba</li> <li>3. Esquema de pruebas unitarias</li> </ol>

**Tabla LII. Sesión 15: presentaciones: “herramientas para pruebas”**

<b>SESIÓN 15:</b>	<b>PRESENTACIONES: “HERRAMIENTAS PARA PRUEBAS”</b>
<b>OBJETIVO:</b>	<b>Conocer las herramientas para realizar pruebas unitarias, de stress y de rendimiento para las tecnologías más utilizadas en el mercado.</b>
<b>CONTENIDO:</b>	<p>Se realizarán presentaciones en grupos sobre diversas herramientas para: Pruebas unitarias Pruebas de carga para aplicaciones web</p> <p>Los temas de exposición serán asignados a los grupos de laboratorio por el auxiliar de cátedra.</p> <p>Cada grupo deberá de realizar un trabajo escrito que será entregado al auxiliar de cátedra y a los alumnos del curso.</p> <p>Los grupos deberán de ser formados acorde al número de estudiantes del curso, y pueden ser los mismos grupos que se formaron para el proyecto de clase.</p>

**Tabla LIII. Sesión 16: integración de sistemas**

<b>SESIÓN 16:</b>	<b>INTEGRACIÓN DE SISTEMAS</b>
<b>OBJETIVO:</b>	<b>Conocer algunos métodos utilizados actualmente para la integración de sistemas.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Métodos de integración de sistemas 1             <ol style="list-style-type: none"> <li>1.1. Integración de sistemas (definición)</li> <li>1.2. Colas de mensajes                 <ol style="list-style-type: none"> <li>1.2.1. Definición</li> <li>1.2.2. Implementación</li> <li>1.2.3. Algunas alternativas                     <ol style="list-style-type: none"> <li>1.2.3.1. Productos propietarios                         <ol style="list-style-type: none"> <li>1.2.3.1.1. Microsoft Message Queue</li> <li>1.2.3.1.2. MQ (IBM)</li> </ol> </li> <li>1.2.3.2. <i>Open source</i> (JMS implementado en JBoss)</li> </ol> </li> <li>1.2.4. Ventajas</li> <li>1.2.5. Desventajas</li> </ol> </li> <li>1.3. <i>Web Services</i> <ol style="list-style-type: none"> <li>1.3.1. Definición</li> <li>1.3.2. Implementación</li> <li>1.3.3. Ventajas</li> <li>1.3.4. Desventajas</li> </ol> </li> </ol> </li> </ol>

**Tabla LIV. Sesión 17: *Message Broker***

<b>SESIÓN 17:</b>	<b>MESSAGE BROKER</b>
<b>OBJETIVO:</b>	<b>Conocer las funciones básicas de un ESB y algunos productos que lo implementan</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Métodos de integración de sistemas 2             <ol style="list-style-type: none"> <li>1.1. <i>Message Broker</i> <ol style="list-style-type: none"> <li>1.1.1. Definición</li> <li>1.1.2. Funciones principales</li> </ol> </li> <li>1.2. ESB                 <ol style="list-style-type: none"> <li>1.2.1. Definición</li> <li>1.2.2. Funciones principales</li> <li>1.2.3. Características</li> <li>1.2.4. Productos                     <ol style="list-style-type: none"> <li>1.2.4.1. Propietarios                         <ol style="list-style-type: none"> <li>1.2.4.1.1. WebSphere</li> <li>1.2.4.1.2. Bistalk Server</li> </ol> </li> <li>1.2.4.2. <i>Open source</i> <ol style="list-style-type: none"> <li>1.2.4.2.1. Mule</li> <li>1.2.4.2.2. Open ESB</li> <li>1.2.4.2.3. JBoss</li> </ol> </li> </ol> </li> </ol> </li> </ol> </li> </ol>

**Tabla LV. Sesión 18: herramienta *Message Broker***

<b>SESIÓN 18:</b>	<b>HERRAMIENTA <i>MESSAGE BROKER</i></b>
<b>OBJETIVO:</b>	<b>Capacitar al estudiante para configurar un ESB en particular</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"><li>1. ESB Mule<ol style="list-style-type: none"><li>1.1. Breve descripción</li><li>1.2. Prerrequisitos para instalación</li><li>1.3. Configuración del entorno Java</li></ol></li><li>2. Redireccionar peticiones a servicios web externos desde Mule<ol style="list-style-type: none"><li>2.1. Archivo de configuración</li></ol></li></ol>
<b>PARTE PRÁCTICA:</b>	El auxiliar deberá mostrar de forma práctica la forma como se configura un ESB en particular para el redireccionamiento de paquetes. Como sugerencia puede utilizarse el ESB Mule.

**Tabla LVI. Sesión 19: sesión extra**

<b>SESIÓN 19:</b>	<b>SESIÓN EXTRA</b>
<b>OBJETIVO:</b>	<b>Esta sesión se deja para reponer alguna sesión que no se haya podido impartir por algún motivo en particular.</b>

**Tabla LVII. Sesión 20: evaluación final**

<b>SESIÓN 20:</b>	<b>EVALUACIÓN FINAL</b>
<b>OBJETIVO:</b>	Durante esta sesión, se realizará la evaluación final del laboratorio.

#### 4.6 Detalle de tareas asignadas

No aplica para este laboratorio.

#### 4.7 Detalle de tutoriales

**Tabla LVIII. Tutorial 1**

<b>TUTORIAL:</b>	<b>Tutorial 1</b>
<b>OBJETIVO:</b>	Elaborar un plan para administración de la configuración.
Iniciado en la sesión de clase número:	1
Completado en las sesión de clase número:	3
Demo a ser conducido:	No aplica
Criterio de evaluación:	Nomenclatura del sistema de versiones .....50% Utilidad del sistema de versiones para Acoplarlo al proyecto de clase .....20% Presentación completa del documento del plan de administración de la configuración .....30%

Documentación a obtener o resultado esperado:	Un documento que contenga el plan para administración de la configuración
---	---

**Tabla LIX. Tutorial 2**

<b>TUTORIAL:</b>	<b>Tutorial 2</b>
<b>OBJETIVO:</b>	Elaborar diagramas de clases y paquetes para un caso de estudio.
Iniciado en la sesión de clase número:	9
Completado en las sesión de clase número:	9
Demo a ser conducido:	No aplica
Criterio de evaluación:	Conocimiento técnico .....20% Correcta estructuración de elementos de diseño ...20% Correcta definición de clases y paquetes .....30% Análisis de requerimientos no funcionales aplicado a la arquitectura propuesta .....30%
Documentación a obtener o resultado esperado:	Diagrama de paquetes y casos de uso utilizando UML

**Tabla LX. Tutorial 3**

<b>TUTORIAL:</b>	<b>Tutorial 3</b>
<b>OBJETIVO:</b>	Elaborar un esquema de pruebas unitarias y de estrés.
Iniciado en la sesión de clase número:	13
Completado en las sesión de clase número:	13

Demo a ser conducido:	No Aplica
Criterio de evaluación:	Correcta definición de pruebas unitarias .....40% Correcta definición de pruebas de integración .....40% Selección de la herramientas para pruebas .....20%
Documentación a obtener o resultado esperado:	Un documento que contenga el esquema de pruebas unitarias y de integración.

## 4.8 Detalle de evaluaciones

**Tabla LXI. Evaluaciones**

<b>NÚMERO TOTAL DE EVALUACIONES:</b>	1
<b>EVALUACIÓN NÚMERO:</b>	1
Evaluación a ser tomada después de la sesión número:	19
Temas de la evaluación:	Ingeniería de <i>software</i> II (módulo completo)
Duración de la evaluación:	2 hrs.
Distribución de la evaluación:	Parte teórica - 40 % Parte práctica - 60%

## 4.9 Evaluación de ejemplo

### Parte teórica (40 Pts.):

1. ¿Qué es la administración de la configuración?
2. ¿Qué son los requerimientos no funcionales?
3. ¿Qué es la funcionalidad?
4. ¿Qué es la usabilidad?
5. ¿Qué es el performance?
6. ¿Qué es la programación orientada a aspectos (AOP)?
7. ¿Qué es un servidor de aplicaciones?
8. Mencione el nombre de dos métodos para integración de sistemas

### Parte práctica (60 Pts.):

1. Desarrolle los siguientes temas (10 Pts. c/u)
  - a. *Spring*
  - b. Mapeo objeto-relacional
  - c. Modelo vista controlador
  - d. Métodos de integración de sistemas
2. Justifique por qué seleccionó la herramienta para administración de la configuración que utilizó en su proyecto de clase (10 Pts.)
3. Describa la arquitectura de componentes utilizada en su proyecto de clase y justifique el por qué diseñó dicha estructura. (10 Pts.)

## 4.10 Bibliografía recomendada

1. **Software Quality Management, Courseware.** Tata Consultancy Services, 2006.
2. Crawford, William y Kaplan, Jonathan. **J2EE Design Patterns.** Estados Unidos de Norte América: Editorial O'Reilly, 2003.
3. Freeman, Eric y Freeman, Elisa. **Head First, Design Patterns.** Estados Unidos de Norte América: Editorial O'Reilly, 2004. 638 pp.
4. McLaughlin, Brett y otros. **Head First, Object-Oriented Analysis & Design.** Estados Unidos de Norte América: Editorial O'Reilly, 2006. 600 pp.
5. Johnson, Rod y otros. **Spring java/j2ee Application Framework 2.0.** Año 2007
6. Hohmann, Luke. **Beyond Software Architecture: Creating and Sustaining Winning Solutions.** Estados Unidos de Norte América: Editorial Pearson Education, 2003.
7. Dart, Susan. **Concepts in Configuration Management Systems.** Artículo. Estados Unidos de Norteamérica, Universidad Carnegie Mellon, Software Engineering Institute.
8. [www.hibernate.org](http://www.hibernate.org) (marzo de 2009)
9. [www.sei.cmu.edu](http://www.sei.cmu.edu) (marzo de 2009)
10. <http://maven.apache.org/> (marzo de 2009)
11. <http://subversion.tigris.org/> (marzo de 2009)

12. <http://www.service-architecture.com/object-relational-mapping/articles/> (abril de 2009)
13. [www.ibm.com/websphere](http://www.ibm.com/websphere) (abril de 2009)
14. [www.microsoft.com/biztalk](http://www.microsoft.com/biztalk) (abril de 2009)
15. [www.jboss.org](http://www.jboss.org) (abril de 2009)
16. [www.mulesource.org](http://www.mulesource.org) (julio de 2009)

## **4.11 Anexos**

### **A. Recomendaciones**

1. Es recomendable entregar al principio del semestre la calendarización de actividades (entrega de tareas y presentaciones) a los alumnos del laboratorio para evitar contratiempos posteriores.
2. Es recomendable realizar la asignación de los temas de investigación al inicio del semestre, para evitar contratiempos para la realización de los mismos.
3. Es recomendable explicar a los alumnos al inicio de semestre que actividades se realizarán de forma individual y que otras se realizarán de forma grupal, y cuando este sea el caso, definir desde un inicio el número máximo de integrantes por grupo.
4. Es necesario tener un medio de comunicación con los alumnos, adicional al correo electrónico, por lo que se recomienda registrarse en la Universidad Virtual, o crear un grupo para

mantener la comunicación con los alumnos y publicar documentos.

## **B. Tutoriales**

Los tutoriales son tareas, investigaciones o prácticas que los estudiantes deben de realizar con el fin de complementar el contenido del laboratorio, o bien poner en práctica los conocimientos adquiridos en el mismo.

### **Tutorial 1**

#### **Objetivo**

Elaborar un plan para administración de la configuración (aplicable al proyecto de clase).

#### **Metodología de trabajo**

El estudiante deberá identificar los elementos de la configuración de su proyecto, establecer políticas sobre el manejo de la configuración, y plasmar los resultados en el documento “plan para administración de la configuración”.

#### **Entregable**

El documento “plan para administración de la configuración de la metodología RUP”.

## **Tutorial 2**

### **Objetivo**

Evaluar la capacidad de los estudiantes para diseñar la estructura de componentes para la arquitectura de un sistema.

### **Temas a evaluar**

- Arquitectura del software
  - Componentes
  - Paquetes
  - Clases
- Facilidad de cambio del sistema propuesto

### **Metodología de trabajo**

El estudiante deberá definir la estructura de componentes para un caso de estudio. Se debe de detallar la arquitectura en los niveles de componentes, paquetes y clases.

### **Tutorial 3**

#### **Objetivo**

Definir un esquema de pruebas para el proyecto de clase.

#### **Metodología de trabajo**

El estudiante deberá de seleccionar una herramienta para pruebas unitarias, y realizar el esquema de pruebas unitarias basado en dicha herramienta. Deberá de realizar también la definición de las pruebas de integración (puede utilizar como herramienta de diseño los casos de prueba) y definir la metodología de trabajo y una herramienta de software para las pruebas de *stress* del sistema.

#### **Entregable**

Un documento con la definición de pruebas unitarias y de *stress* aplicable al proyecto de clase. Se debe de seleccionar una herramienta para cada una de las pruebas. Adicionalmente se deben de realizar la especificación para las pruebas de integración del sistema.

#### **Observaciones**

Los tutoriales fueron pensados para dejarse como tarea de casa, por lo que puede realizarse de forma individual, o en grupos según el auxiliar de laboratorio crea conveniente.

El objetivo del tutorial es que el estudiante refuerce sus conocimientos sobre los temas del laboratorio, por lo que el estudiante se encuentra en absoluta libertad de solventar dudas sobre los tutoriales con el auxiliar del curso.

## 5. GUÍA DEL INSTRUCTOR DE *SOFTWARE* AVANZADO

A continuación se presenta el desarrollo de la guía del curso de *Software Avanzado*, de acuerdo al formato sugerido en el curso de “Estructuración de laboratorios”, impartido por el señor. Mrutunjaya Panda.

### 5.1 Información general

NOMBRE DE CURSO: *Software Avanzado* (780)

PRE- REQUISITOS: *Análisis y Diseño de Sistemas 2* (785)

POST – REQUISITOS: Ninguno

OBJETIVO: Que el estudiante conozca los conceptos básicos relacionados con la administración de proyectos incluyendo calendarización, análisis de riesgos, cálculo del retorno de inversión a través de las metodologías propuestas por Microsoft *Solution Framework*, TPS y PSP

MÓDULOS: Administración de Proyectos

NOTA DE PROMOCIÓN: 61 puntos

## 5.2 Distribución de horas y actividades

**Tabla LXII. Distribución del curso *software* avanzado**

<b>MODULO</b>	<b>TEORÍA</b>	<b>PRÁCTICA</b>	<b>TOTAL</b>	<b>EXAMEN</b>	<b>TUTORIALES</b>
ADMINISTRACIÓN DE PROYECTOS	28 horas	10 horas	38 horas	2 horas	5
<b>TOTAL</b>	<b>34 horas</b>	<b>4 horas</b>	<b>38 horas</b>	<b>2 horas</b>	<b>5</b>

## 5.3 Evaluación

Ponderación de tareas, prácticas y proyectos.

**Tabla LXIII. Evaluación del curso de *software* avanzado**

ACTIVIDAD	NUMERO	PONDERACIÓN	TOTAL
ASISTENCIAS O EXAMENES CORTOS	10	1.00 %	10.00 %
TUTORIALES	5	15.00 %	75.00 %
EVALUACIÓN	1	15.00 %	15.00 %
		<b>TOTAL</b>	<b>100.00 %</b>

#### 5.4 Detalle de sesiones

**Tabla LXIV. Detalle de sesiones del curso de *software* avanzado**

NO	NOMBRE DE SESIÓN	TIPO DE SESIÓN		TIPO DE ACTIVIDAD
		TEÓRICA	PRÁCTICA	
1	MICROSOFT SOLUTIONS FRAMEWORK	X		
2	MODELO DE EQUIPO	X		
3	MODELO DE PROCESO	X		
4	DISCIPLINA DE GESTIÓN DE RIESGO	X		
5	ANÁLISIS DE RIESGO		X	TUTORIAL
6	GESTIÓN DE PROYECTOS	X		
7	ESTIMACIÓN	X		
8	PUNTOS DE FUNCIÓN		X	TUTORIAL
9	PLANIFICACIÓN		X	TUTORIAL
10	MODELO DE COSTO TOTAL DE PROPIEDAD	X		
11	RETORNO DE INVERSIÓN	X		
12	RDI PARA INSPECCIÓN DE SOFTWARE		X	TUTORIAL
13	PRESUPUESTACION	X		
14	MODELO DISEÑO DE LA SOLUCIÓN	X		
15	DESARROLLO Y ESTABILIZACIÓN	X		
16	TSP	X		
17	PSP	X		
18	ADMINISTRACIÓN DE PERSONAL CON PSP	X		

19	TUTORIAL PSP Y TSP		X	TUTORIAL
20	SESIÓN EXTRA			
21	EVALUACIÓN FINAL		X	EVALUACIÓN

## 5.5 Distribución de sesiones

**Tabla LXV. Sesión 1: Microsoft *Solutions Framework***

<b>SESIÓN 1:</b>	<b>MICROSOFT SOLUTIONS FRAMEWORK</b>
<b>OBJETIVO:</b>	<b>Introducir al estudiante en el marco para gestión de proyectos Microsoft <i>Solutions Framework</i>.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Microsoft <i>Solutions Framework</i> <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. Principios básicos</li> </ol> </li> <li>2. Modelo de equipo           <ol style="list-style-type: none"> <li>2.1. Roles en el modelo de equipo</li> <li>2.2. Beneficios</li> </ol> </li> <li>3. Modelo de proceso           <ol style="list-style-type: none"> <li>3.1. Definición</li> <li>3.2. Fases</li> <li>3.3. Beneficios</li> </ol> </li> <li>4. Modelo de diseño de la solución           <ol style="list-style-type: none"> <li>4.1. Definición</li> <li>4.2. Tres perspectivas del modelo</li> <li>4.3. Beneficios</li> </ol> </li> <li>5. Modelo de aplicación           <ol style="list-style-type: none"> <li>5.1. Definición</li> <li>5.2. Beneficios</li> </ol> </li> <li>6. Modelo de la arquitectura empresarial           <ol style="list-style-type: none"> <li>6.1. Definición</li> <li>6.2. Perspectivas del modelo               <ol style="list-style-type: none"> <li>6.2.1. Arquitectura del negocio</li> <li>6.2.2. Arquitectura de la aplicación</li> <li>6.2.3. Arquitectura de la información</li> <li>6.2.4. Arquitectura de la tecnología</li> </ol> </li> <li>6.3. Beneficios</li> </ol> </li> <li>7. Modelo de infraestructura           <ol style="list-style-type: none"> <li>7.1. Definición</li> <li>7.2. Beneficios</li> </ol> </li> <li>8. Modelo de costo total de propiedad           <ol style="list-style-type: none"> <li>8.1. Definición</li> <li>8.2. Beneficios</li> </ol> </li> </ol>

**Tabla LXVI. Sesión 2: modelo de equipo**

<b>SESIÓN 2:</b>	<b>MODELO DE EQUIPO</b>
<b>OBJETIVO:</b>	<b>Comprender las funciones de cada rol dentro del modelo de equipo y algunos aspectos administrativos del modelo.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Modelo de equipo               <ol style="list-style-type: none"> <li>1.1. Definición                   <ol style="list-style-type: none"> <li>1.1.1. Fundamentos                       <ol style="list-style-type: none"> <li>1.1.1.1. Relacionar roles para apoyar grupos y áreas funcionales</li> <li>1.1.1.2. Definir y diseñar la solución con todos los roles representados</li> <li>1.1.1.3. El modelo de equipo no es un cuadro organizacional</li> <li>1.1.1.4. Grupos pequeños, trabajo en paralelo con puntos de sincronización frecuentes</li> </ol> </li> <li>1.2. Principios                       <ol style="list-style-type: none"> <li>1.2.1. Contabilidad clara, responsabilidad compartida</li> <li>1.2.2. Dar poder a los miembros del equipo</li> <li>1.2.3. Enfoque en el valor del negocio</li> <li>1.2.4. Comunicación abierta</li> </ol> </li> <li>1.3. Roles en el modelo de equipo                       <ol style="list-style-type: none"> <li>1.3.1. Administrador del producto                           <ol style="list-style-type: none"> <li>1.3.1.1. Definición</li> <li>1.3.1.2. Áreas funcionales</li> </ol> </li> <li>1.3.2. Administrador del programa                           <ol style="list-style-type: none"> <li>1.3.2.1. Definición</li> <li>1.3.2.2. Áreas funcionales</li> </ol> </li> <li>1.3.3. Desarrollador                           <ol style="list-style-type: none"> <li>1.3.3.1. Definición</li> <li>1.3.3.2. Áreas funcionales</li> </ol> </li> <li>1.3.4. Pruebas                           <ol style="list-style-type: none"> <li>1.3.4.1. Definición</li> <li>1.3.4.2. Áreas funcionales</li> </ol> </li> <li>1.3.5. Capacitador de usuarios                           <ol style="list-style-type: none"> <li>1.3.5.1. Definición</li> <li>1.3.5.2. Áreas funcionales</li> </ol> </li> <li>1.3.6. Logística                           <ol style="list-style-type: none"> <li>1.3.6.1. Definición</li> </ol> </li> <li>1.3.7. Tabla comparativa</li> </ol> </li> </ol> </li> </ol> </li> </ol>

**Tabla LXVII. Sesión 3: modelo de proceso**

<b>SESIÓN 3:</b>	<b>MODELO DE PROCESO</b>
<b>OBJETIVO:</b>	<b>Que el estudiante conozca los elementos principales del las fases de visionadao y planificación del modelo de proceso de Microsoft <i>Solutions Framework</i>.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Modelo de proceso               <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. Fases</li> </ol> </li> <li>2. Fase de visionado               <ol style="list-style-type: none"> <li>2.1. Definición</li> <li>2.2. Declaración del problema</li> <li>2.3. Declaración de la visión</li> <li>2.4. Alcance</li> <li>2.5. Tareas asignadas a los distintos roles del proyecto durante la fase de visionado</li> </ol> </li> <li>3. Fase de planificación               <ol style="list-style-type: none"> <li>3.1. Análisis de los requerimientos del negocio</li> <li>3.2. Retorno de inversión</li> <li>3.3. Modelo de costo total de propiedad</li> <li>3.4. Administración del riesgo</li> <li>3.5. Análisis funcional                   <ol style="list-style-type: none"> <li>3.5.1. Requerimientos de seguridad</li> <li>3.5.2. Requerimientos del proceso de flujo de trabajo</li> <li>3.5.3. Requerimientos de rendimiento</li> <li>3.5.4. Requerimientos de mantenibilidad</li> <li>3.5.5. Requerimientos de extensibilidad</li> <li>3.5.6. Requerimientos de disponibilidad</li> <li>3.5.7. Requerimientos del factor humano</li> <li>3.5.8. Requerimientos de escalabilidad</li> </ol> </li> <li>3.6. Establecer el cronograma del proyecto</li> <li>3.7. Definición de roles del proyecto durante la fase de planificación</li> </ol> </li> </ol>

**Tabla LXVIII. Sesión 4: disciplina de gestión de riesgo**

<b>SESIÓN 4:</b>	<b>DISCIPLINA DE GESTIÓN DE RIESGO</b>
<b>OBJETIVO:</b>	<b>Que el estudiante conozca los aspectos fundamentales de la disciplina de Gestión de Riesgos de Microsoft <i>Solutions Framework</i></b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Disciplina de gestión del riesgo               <ol style="list-style-type: none"> <li>1.1. Atributos</li> </ol> </li> <li>2. Definición de riesgo</li> <li>3. Principios que fundamentan esta disciplina</li> <li>4. Integrar el análisis de riesgo al ciclo de vida de un proyecto</li> <li>5. Proceso de gestión del riesgo               <ol style="list-style-type: none"> <li>5.1. Pasos del proceso de gestión del riesgo con MSF                   <ol style="list-style-type: none"> <li>5.1.1. Identificar riesgos                       <ol style="list-style-type: none"> <li>5.1.1.1. Metas</li> <li>5.1.1.2. Clasificación de riesgos</li> <li>5.1.1.3. Declaraciones de riesgos</li> <li>5.1.1.4. Resultados</li> </ol> </li> <li>5.1.2. Analizar y priorizar riesgos                       <ol style="list-style-type: none"> <li>5.1.2.1. Metas</li> <li>5.1.2.2. Probabilidad del riesgo</li> <li>5.1.2.3. Impacto del riesgo</li> <li>5.1.2.4. Priorización de actividades de riesgo</li> <li>5.1.2.5. Resultados</li> </ol> </li> <li>5.1.3. Planificar y calendarizar el manejo del riesgo                       <ol style="list-style-type: none"> <li>5.1.3.1. Metas</li> <li>5.1.3.2. Planificación de acciones de riesgo</li> <li>5.1.3.3. Resultados</li> </ol> </li> <li>5.1.4. Seguimiento y reporte del estado del riesgo                       <ol style="list-style-type: none"> <li>5.1.4.1. Metas</li> <li>5.1.4.2. Actividades de seguimiento</li> <li>5.1.4.3. Actividades de reporte</li> <li>5.1.4.4. Resultados</li> </ol> </li> <li>5.1.5. Control de riesgo                       <ol style="list-style-type: none"> <li>5.1.5.1. Metas</li> <li>5.1.5.2. Actividades de control de riesgos</li> <li>5.1.5.3. Resultados</li> </ol> </li> <li>5.1.6. Aprender del riesgo                       <ol style="list-style-type: none"> <li>5.1.6.1. Metas</li> <li>5.1.6.2. Capturar el conocimiento acerca del riesgo</li> <li>5.1.6.3. Gestionando el conocimiento de riesgos</li> <li>5.1.6.4. Retener el conocimiento de riesgos</li> <li>5.1.6.5. Resultados</li> </ol> </li> </ol> </li> </ol> </li> </ol>

**Tabla LXIX. Sesión 5: análisis de riesgo**

<b>SESIÓN 5:</b>	<b>ANÁLISIS DE RIESGO (SESIÓN PRÁCTICA)</b>
<b>OBJETIVO:</b>	<b>Aplicar los conocimientos adquiridos durante la sesión de análisis de riesgo mediante un ejercicio práctico</b>
<b>CONTENIDO:</b>	<p>Durante esta sesión, el estudiante deberá hacer el análisis de riesgo y presentar como entregable la <b>plantilla para análisis de riesgo</b> (Risk Template Tool.xls) para un caso de estudio en particular.</p> <p>El trabajo será realizado en grupos con un número de estudiantes definidos por el auxiliar de cátedra.</p> <p><b>Nota:</b> De no completarse la plantilla, la misma se entregará en el siguiente laboratorio. Cualquier duda para la realización de estimaciones y sobre los diversos puntos de la plantilla se resolverán en clase.</p>

**Tabla LXX. Sesión 6: gestión de proyectos**

<b>SESIÓN 6:</b>	<b>GESTIÓN DE PROYECTOS</b>
<b>OBJETIVO:</b>	<b>Conocer los aspectos más relevantes de la disciplina de gestión de proyectos de Microsoft <i>Solutions Framework</i></b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. ¿Qué es gestión de proyectos?             <ol style="list-style-type: none"> <li>1.1. Áreas de la gestión de proyectos                 <ol style="list-style-type: none"> <li>1.1.1. Planificación de proyectos, seguimiento y control de cambios</li> <li>1.1.2. Gestión del alcance</li> <li>1.1.3. Gestión de cronogramas</li> <li>1.1.4. Gestión de costos</li> <li>1.1.5. Gestión de los recursos de personal</li> <li>1.1.6. Gestión de la comunicación</li> <li>1.1.7. Administración del riesgo</li> <li>1.1.8. Obtenciones</li> <li>1.1.9. Gestión de la calidad</li> </ol> </li> </ol> </li> <li>2. Gestión de proyectos y los procesos específicos de TI</li> <li>3. Características de la gestión de proyectos de MSF             <ol style="list-style-type: none"> <li>3.1. El Rol de administrador de proyectos está inmerso en la gestión de proyectos</li> <li>3.2. El gestor de proyectos y los líderes de equipos</li> <li>3.3. Responsabilidades de la gestión de proyectos</li> </ol> </li> <li>4. Recomendaciones para los equipos             <ol style="list-style-type: none"> <li>4.1. Administración del alcance</li> <li>4.2. Preparar planes</li> <li>4.3. Reutilización de documentos</li> <li>4.4. Planes de proyecto</li> <li>4.5. Estructura de desglose de trabajo</li> <li>4.6. Estimación en proyectos de software</li> <li>4.7. Recomendaciones de calendarización</li> </ol> </li> </ol>

**Tabla LXXI. Sesión 7: estimación**

SESIÓN 7:	ESTIMACIÓN
<b>OBJECTIVO:</b>	<b>Que el estudiante conozca algunas técnicas útiles para la estimación de tiempos de desarrollo de un proyecto.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Definición de estimación</li> <li>2. Técnicas para estimación               <ol style="list-style-type: none"> <li>2.1. Técnicas de descomposición                   <ol style="list-style-type: none"> <li>2.1.1. Líneas de código                       <ol style="list-style-type: none"> <li>2.1.1.1. Ejemplo de líneas de código</li> </ol> </li> <li>2.1.2. Estimación por puntos de función                       <ol style="list-style-type: none"> <li>2.1.2.1. Ejemplo</li> </ol> </li> </ol> </li> <li>2.2. Estimación empírica                   <ol style="list-style-type: none"> <li>2.2.1. Modelos estáticos de variable simple                       <ol style="list-style-type: none"> <li>2.2.1.1. COCOMO</li> </ol> </li> <li>2.2.2. Modelos estáticos de multivariable</li> <li>2.2.3. Modelos dinámicos de multivariable</li> </ol> </li> </ol> </li> </ol>
<b>NOTA:</b>	Al finalizar la presentación, el auxiliar de laboratorio deberá mostrar con un ejemplo la forma como se calculan los puntos de función.

**Tabla LXXII. Sesión 8: puntos de función**

SESIÓN 8:	PUNTOS DE FUNCIÓN (SESIÓN PRÁCTICA)
<b>OBJECTIVO:</b>	<b>Que el estudiante ponga en práctica los conocimientos adquiridos sobre las técnicas de estimación mediante un ejercicio práctico.</b>
<b>CONTENIDO:</b>	<p>Los estudiantes realizarán la estimación por puntos de función para un caso de estudio en particular.            El trabajo será realizado en grupos con un número de estudiantes definidos por el auxiliar de cátedra.  <b>Nota:</b> De no completarse la el documento, la misma se entregará en el siguiente laboratorio. Cualquier duda para la realización de estimaciones y sobre los diversos puntos de la plantilla se resolverán en clase.</p>

**Tabla LXXIII. Sesión 9: planificación**

<b>SESIÓN 9:</b>	<b>PLANIFICACIÓN (SESIÓN PRÁCTICA)</b>
<b>OBJETIVO:</b>	<b>Que el estudiante ponga en práctica los conocimientos adquiridos sobre la disciplina de planificación de proyectos</b>
<b>CONTENIDO:</b>	<p>Durante esta sesión, el estudiante deberá realizar el análisis de tiempos y del proyecto presentando como entregable la <b>plantilla de estructura del proyecto</b> (Project Structure.doc) para un caso de estudio en particular.</p> <p>El trabajo será realizado en grupos con un número de estudiantes definidos por el auxiliar de cátedra.</p> <p><b>Nota:</b> De no completarse la plantilla, la misma se entregará en el siguiente laboratorio. Cualquier duda para la realización de estimaciones y sobre los diversos puntos de la plantilla se resolverán en clase.</p>

**Tabla LXXIV. Sesión 10: modelo de costo total de propiedad**

<b>SESIÓN 10:</b>	<b>MODELO DE COSTO TOTAL DE PROPIEDAD</b>
<b>OBJETIVO:</b>	<b>Que el estudiante comprenda los aspectos más importantes del modelo de costo total de propiedad de MSF</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Modelo de costo total de propiedad               <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. Beneficios</li> <li>1.3. Costos</li> <li>1.4. Retorno de inversión</li> <li>1.5. Consideraciones a tomar en cuenta</li> <li>1.6. Ciclo de vida del costo total de propiedad                   <ol style="list-style-type: none"> <li>1.6.1. Fase de análisis</li> <li>1.6.2. Fase de mejora del costo total de propiedad</li> <li>1.6.3. Fase de administración del costo total de propiedad</li> </ol> </li> </ol> </li> <li>2. Como minimizar el costo total de propiedad</li> <li>3. Análisis costo beneficio               <ol style="list-style-type: none"> <li>3.1. Definición de costos</li> <li>3.2. Definición de beneficios</li> <li>3.3. Categorías de costos y beneficios</li> <li>3.4. Categorías de costos                   <ol style="list-style-type: none"> <li>3.4.1. Costos de desarrollo</li> <li>3.4.2. Costos de operación</li> <li>3.4.3. Otras clasificaciones                       <ol style="list-style-type: none"> <li>3.4.3.1. Costos de <i>hardware</i> y <i>software</i></li> <li>3.4.3.2. Costos de personal</li> <li>3.4.3.3. Costos de instalación</li> <li>3.4.3.4. Costos operativos</li> <li>3.4.3.5. Costos de suministros</li> </ol> </li> </ol> </li> </ol> </li> </ol>

**Tabla LXXV. Sesión 11: retorno de inversión**

<b>SESIÓN 11:</b>	<b>RETORNO DE INVERSIÓN</b>
<b>OBJETIVO:</b>	<b>Que el estudiante conozca algunos de los métodos principales para el cálculo del retorno de inversión</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Definición de retorno de inversión</li> <li>2. Métodos para análisis del retorno de la inversión               <ol style="list-style-type: none"> <li>2.1. Beneficio</li> <li>2.2. Costo</li> <li>2.3. Razón costo-beneficio</li> <li>2.4. Retorno de la inversión</li> <li>2.5. Valor actual neto</li> <li>2.6. Punto de equilibrio</li> </ol> </li> </ol>

**Tabla LXXVI. Sesión 12: RDI para inspección de *software***

<b>SESIÓN 12:</b>	<b>RDI PARA INSPECCIÓN DE SOFTWARE</b>
<b>OBJETIVO:</b>	<b>Que el estudiante conozca algunas metodologías de retorno de la inversión para el proceso de inspección de <i>software</i></b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Metodología RDI para el proceso de inspección de <i>software</i> <ol style="list-style-type: none"> <li>1.1. Metodología de inspección de costos</li> <li>1.2. Metodología de inspección beneficio</li> <li>1.3. Metodología de inspección razón beneficio/costo</li> <li>1.4. Metodología de inspección RDI%</li> <li>1.5. Metodología de inspección del valor actual neto</li> <li>1.6. Metodología de inspección del punto de equilibrio</li> </ol> </li> </ol>

**Tabla LXXVII. Sesión 13: presupuestación**

<b>SESIÓN 13:</b>	<b>PRESUPUESTACION (SESIÓN PRÁCTICA)</b>
<b>OBJETIVO:</b>	<b>Que el estudiante ponga en práctica los conocimientos adquiridos sobre análisis y estimación de costos</b>
<b>CONTENIDO:</b>	<p>Durante esta sesión, el estudiante deberá realizar el análisis de costo para un proyecto (caso de estudio) presentando como entregable la <b>plantilla “presupuesto de proyecto”</b> (Budget Plan.doc).</p> <p>El trabajo será realizado en grupos con un número de estudiantes definidos por el auxiliar de cátedra.</p> <p><b>Nota:</b> De no completarse la plantilla se, la misma se entregará en el siguiente laboratorio. Cualquier duda para la realización de estimaciones y sobre los diversos puntos de la plantilla se resolverán en clase.</p>

**Tabla LXXVIII. Sesión 14: modelo diseño de la solución**

<b>SESIÓN 14:</b>	<b>MODELO DISEÑO DE LA SOLUCIÓN</b>
<b>OBJETIVO:</b>	<b>Dara a conocer al estudiante los aspectos relevantes de los modelos de diseño de la solución, de aplicación y arquitectura empresarial de MSF.</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Modelo de diseño de la solución             <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. Perspectivas del modelo de diseño de la solución                 <ol style="list-style-type: none"> <li>1.2.1. Diseño conceptual</li> <li>1.2.2. Diseño lógico</li> <li>1.2.3. Diseño físico</li> </ol> </li> </ol> </li> <li>2. Modelo de aplicación             <ol style="list-style-type: none"> <li>2.1. Definición</li> <li>2.2. Modelo de servicios                 <ol style="list-style-type: none"> <li>2.2.1. Servicios de usuario</li> <li>2.2.2. Servicios del negocio</li> <li>2.2.3. Servicios de datos</li> </ol> </li> <li>2.3. Arquitectura en capas                 <ol style="list-style-type: none"> <li>2.3.1. Una capa</li> <li>2.3.2. Dos capas</li> <li>2.3.3. Tres capas</li> <li>2.3.4. N-capas</li> </ol> </li> </ol> </li> <li>3. Modelo de la arquitectura empresarial             <ol style="list-style-type: none"> <li>3.1. Arquitectura de tecnología                 <ol style="list-style-type: none"> <li>3.1.1. Definición</li> <li>3.1.2. Fundamentos de redes</li> <li>3.1.3. Roles en las redes de computadoras</li> <li>3.1.4. Servidores de servicios</li> <li>3.1.5. Tipos de redes</li> <li>3.1.6. Intercambio electrónico de datos</li> <li>3.1.7. Tipos de solución</li> </ol> </li> </ol> </li> </ol>

**Tabla LXXIX. Sesión 15: desarrollo y estabilización**

<b>SESIÓN 15:</b>	<b>DESARROLLO Y ESTABILIZACIÓN</b>
<b>OBJETIVO:</b>	<b>Dar a conocer al estudiante los puntos más importante de las fases de desarrollo y estabilización de sistemas de acuerdo a la metodología MSF</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Fase de desarrollo               <ol style="list-style-type: none"> <li>1.1. Definición</li> <li>1.2. Definición de roles durante la fase de desarrollo</li> </ol> </li> <li>2. Fase de estabilización               <ol style="list-style-type: none"> <li>2.1. Definición</li> <li>2.2. Definición de roles en la fase de estabilización del <i>software</i></li> </ol> </li> <li>3. Proceso de pruebas               <ol style="list-style-type: none"> <li>3.1. Definición</li> <li>3.2. Tipos de pruebas</li> <li>3.3. Tipos de entregables</li> <li>3.4. Reporte de errores en entregables</li> </ol> </li> <li>4. Despliegue del sistema               <ol style="list-style-type: none"> <li>4.1. Definición</li> <li>4.2. Medios de despliegue</li> <li>4.3. Métodos de despliegue</li> <li>4.4. Despliegue piloto</li> <li>4.5. Despliegue final</li> </ol> </li> </ol>

**Tabla LXXX. Sesión 16: TSP**

<b>SESIÓN 16:</b>	<b>TSP</b>
<b>OBJETIVO:</b>	<b>Introducir al estudiante en los conceptos de administración de equipos mediante la metodología de trabajo TSP</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. <i>Team Software Process</i> (TSP)               <ol style="list-style-type: none"> <li>1.1. Definición</li> </ol> </li> <li>2. Desarrollo de TSP               <ol style="list-style-type: none"> <li>2.1. Ingeniería de los equipos de trabajo</li> <li>2.2. Las condiciones de los equipos de trabajo</li> <li>2.3. Equipos efectivos</li> <li>2.4. Construyendo equipos efectivos</li> </ol> </li> <li>3. Un proceso de equipo operacional</li> <li>4. La estructura de TSP</li> <li>5. Lanzamiento de un equipo TSP</li> <li>6. El proceso de trabajo en equipo de TSP               <ol style="list-style-type: none"> <li>6.1. Liderando el equipo</li> <li>6.2. Comunicación</li> <li>6.3. Mantenimiento del plan</li> <li>6.4. Rebalanceo de la carga de trabajo del equipo</li> </ol> </li> <li>7. Administración de la calidad con TSP               <ol style="list-style-type: none"> <li>7.1. El plan de la calidad</li> <li>7.2. Identificar los problemas de calidad</li> </ol> </li> </ol>

	7.3. Buscar y prevenir problemas de calidad
--	---

**Tabla LXXXI. Sesión 17: PSP**

<b>SESIÓN 17:</b>	<b>PSP</b>
<b>OBJETIVO:</b>	<b>Que el estudiante conozca la metodología propuesta por PSP para administración de personal</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. PSP             <ol style="list-style-type: none"> <li>1.1. Definición</li> </ol> </li> <li>2. Principios</li> <li>3. Estructura de proceso             <ol style="list-style-type: none"> <li>3.1. Planificación</li> <li>3.2. Estimación de tamaño</li> <li>3.3. Cálculo</li> <li>3.4. Estimación de recursos</li> </ol> </li> </ol>

**Tabla LXXXII. Sesión 18: administración de personal con PSP**

<b>SESIÓN 18:</b>	<b>ADMINISTRACIÓN DE PERSONAL CON PSP</b>
<b>OBJETIVO:</b>	<b>Que el estudiante conozca la metodología propuesta por PSP para administración de personal</b>
<b>CONTENIDO:</b>	<ol style="list-style-type: none"> <li>1. Obtención de datos             <ol style="list-style-type: none"> <li>1.1. Medición de tiempos</li> <li>1.2. Tamaño de las mediciones                 <ol style="list-style-type: none"> <li>1.2.1. Líneas de código</li> <li>1.2.2. Categorías de tamaño</li> <li>1.2.3. Contabilidad de tamaño</li> </ol> </li> <li>1.3. Mediciones de calidad</li> </ol> </li> <li>2. Administración de la calidad             <ol style="list-style-type: none"> <li>2.1. Defectos y calidad</li> <li>2.2. La Responsabilidad del ingeniero</li> <li>2.3. Remoción temprana de defectos</li> <li>2.4. Prevención de defectos</li> </ol> </li> </ol>

**Tabla LXXXIII. Sesión 19: tutorial PSP y TSP**

<b>SESIÓN 19:</b>	<b>TUTORIAL PSP Y TSP</b>
<b>OBJETIVO:</b>	<b>Que el estudiante pueda inferir el comportamiento de un equipo utilizando las metodologías PSP y TSP</b>
<b>CONTENIDO:</b>	Durante esta sesión, el estudiante deberá realizar el análisis del comportamiento de un equipo de trabajo mediante la interpretación de un conjunto de mediciones de un caso de estudio en particular. El trabajo será realizado en grupos con un número de estudiantes definidos por el auxiliar de cátedra. <b>Nota:</b> De no completarse la plantilla, la misma se entregará en el siguiente laboratorio. Cualquier duda para la realización de estimaciones y sobre los diversos puntos de la plantilla se resolverán en clase.

**Tabla LXXXIV. Sesión 20: sesión extra**

<b>SESIÓN 20:</b>	<b>SESIÓN EXTRA</b>
<b>OBJETIVO:</b>	<b>Esta sesión se deja para reponer alguna sesión que no se haya podido impartir por algún motivo en particular.</b>

**Tabla LXXXV. Sesión 21: evaluación final**

<b>SESIÓN 21:</b>	<b>EVALUACIÓN FINAL</b>
<b>OBJETIVO:</b>	<b>Durante esta sesión, se realizará la evaluación final del laboratorio.</b>

## **5.6 Detalle de tareas asignadas**

No aplica para este laboratorio.

## 5.7 Detalle de tutoriales

**Tabla LXXXVI. Tutorial 1**

<b>TUTORIAL:</b>	<b>Tutorial 1</b>
<b>OBJETIVO:</b>	Que el estudiante ponga en práctica los conocimientos adquiridos durante la sesión de análisis de riesgo
Iniciado en la sesión de clase número:	4
Completado en las sesión de clase número:	4
Demo a ser conducido:	No aplica
Criterio de evaluación:	Identificación del riesgo ..... 30% Análisis y priorización de riesgo..... 35% Plan de riesgos ..... 35%
Documentación a obtener o resultado esperado:	Documento de análisis de riesgo

**Tabla LXXXVII. Tutorial 2**

<b>TUTORIAL:</b>	<b>Tutorial 2</b>
<b>OBJETIVO:</b>	Que el estudiante ponga en práctica los conocimientos adquiridos sobre las técnicas de estimación mediante un ejercicio práctico.
Iniciado en la sesión de clase número:	7
Completado en las sesión de clase número:	7
Demo a ser conducido:	No aplica

Criterio de evaluación:	Calculo de FP..... 50% Estimación de tiempos..... 50%
Documentación a obtener o resultado esperado:	Documentación del trabajo realizado

**Tabla LXXXVIII. Tutorial 3**

<b>TUTORIAL:</b>	<b>Tutorial 3</b>
<b>OBJETIVO:</b>	Que el estudiante ponga en práctica los conocimientos adquiridos sobre la disciplina de planificación de proyectos
Iniciado en la sesión de clase número:	6
Completado en las sesión de clase número:	6
Demo a ser conducido:	No aplica
Criterio de evaluación:	Metas, objetivos, asunciones y restricciones.....5% Aproximación maestra del proyecto.....15% Aproximación de hitos.....10% Estimación de proyectos.....20% Cronograma.....20% Roles y responsabilidades.....10% Protocolos del proyecto.....10% Riesgos.....10%
Documentación a obtener o resultado esperado:	Documento de estructura del proyecto

**Tabla LXXXIX. Tutorial 4**

<b>TUTORIAL:</b>	<b>Tutorial 4</b>
<b>OBJETIVO:</b>	Que el estudiante ponga en práctica los conocimientos adquiridos sobre análisis y estimación de costos.
Iniciado en la sesión de clase número:	10
Completado en las sesión de clase número:	11
Demo a ser conducido:	No aplica
Criterio de evaluación:	Estimación de proyecto.....25% Detalles de proyecto.....50% Áreas de costo.....25%
Documentación a obtener o resultado esperado:	Documento "presupuesto del proyecto"

**Tabla XC. Tutorial 5**

<b>TUTORIAL:</b>	<b>Tutorial 5</b>
<b>OBJETIVO:</b>	Que el estudiante pueda inferir el comportamiento de un equipo utilizando las metodologías PSP y TSP
Iniciado en la sesión de clase número:	16
Completado en las sesión de clase número:	18
Demo a ser conducido:	No aplica
Criterio de evaluación:	Queda a discreción del auxiliar de laboratorio

Documentación a obtener o resultado esperado:	Documentación del trabajo realizado
---	-------------------------------------

## 5.8 Detalle de evaluaciones

**Tabla XCI. Evaluaciones**

<b>NÚMERO TOTAL DE EVALUACIONES:</b>	1
<b>EVALUACIÓN NÚMERO:</b>	1
Evaluación a ser tomada después de la sesión número:	20
Temas de la evaluación:	Administración de proyectos, módulo completo
Duración de la evaluación:	2 hrs.
Distribución de la evaluación:	Teoría - 35 % Práctica - 70%

## 5.9 Evaluación de ejemplo

### Parte teórica (40 Pts.):

1. Es un conjunto de buenas prácticas propuesta por Microsoft para la planificación, desarrollo y despliegue de los productos de *software*:
2. ¿Cuáles son los tres servicios del modelo de aplicación de MSF?
3. La posibilidad de sufrir un fallo es conocida como:
4. Este modelo de MSF muestra como estructurar equipos de trabajo para asegurar soluciones de alta calidad:
5. ¿Qué es RDI?
6. Mencione al menos dos tipos de costos en los que se incurre en un proyecto
7. Mencione el nombre de dos métodos para el análisis del RDI
8. ¿Qué es el costo total de propiedad?
9. ¿En qué consiste la estimación de puntos de función?
10. ¿Qué es la administración de la configuración?

### Parte práctica (60 Pts.):

1. Describa las funciones principales de los siguientes roles de un proyecto:
  - a. Administrador del proyecto

- b. Administrador del producto
  - c. Desarrollador
2. Describa cada una de las fases e hitos del modelo de proceso propuesto por MSF
  3. Describa de forma breve en qué consisten cada una de las perspectivas del modelo del diseño de la solución de MSF
  4. Describa en forma breve las siguientes actividades de la gestión de riesgos:
    - a. Identificar riesgos
    - b. Analizar y priorizar riesgos
    - c. Planificar y calendarizar el manejo del riesgo
    - d. Seguimiento y reporte del estado del riesgo
    - e. Control de la mitigación del riesgo y cambio de actividades en el proyecto
  5. Describa en forma breve los principales aspectos de la disciplina de gestión de proyectos de MSF
  6. Describa en forma breve en qué consiste TSP
  7. Describa en forma breve en qué consiste PSP

## 5.10 Bibliografía recomendada

1. **Solution Architecture, Courseware.** Tata Consultancy Services, 2006
2. Turner, Michael S. **Microsoft Solutions Framework Essentials: Building Successful Technology Solutions.** Editorial Microsoft Press, 2006
3. Microsoft Corporation. **Analyzing Requirements and Defining Microsoft .Net Solution Architectures.** Estados Unidos de Norteamérica: Editorial Microsoft Press, 2003.
4. F. Rico, David. **ROI of Software Process Improvement: Metrics for Project Managers and Software Engineers.** Editorial J. Ross Publishing, 2004
5. Wigodsky, Andrew S. **RAPID Value Management for the Business Cost of Ownership: Readiness, Architecture, Process, Integration, Deployment.** Estados Unidos de Norteamérica: Editorial Digital Press, 2004. 297 pp.
6. Humphrey, Watts S. **The Personal Software Process (PSP).** Reporte Técnico. Estados Unidos de Norteamérica, universidad Carnegie Mellon, Software Engineering Institute.
7. Humphrey, Watts S. **The Team Software Process (TSP).** Reporte Técnico. Estados Unidos de Norteamérica, universidad Carnegie Mellon, Software Engineering Institute.
8. Microsoft Corporation. **MSF Risk Management Discipline v.1.1,** White Paper.
9. Microsoft Corporation. **MSF Process Model v.3.1,** White Paper.

10. Microsoft Corporation. **MSF Project Management Discipline v.1.1**, White Paper.
11. Microsoft Corporation. **MSF Team Model v.3.1**, White Paper.
12. Microsoft Corporation. **Microsoft Solutions Framework version 3.0 Overview**, White Paper. Año 2002.

## 5.11 Anexos

### A. Recomendaciones

1. Es recomendable entregar al principio del semestre la calendarización de actividades (entrega de tareas y presentaciones) a los alumnos del laboratorio para evitar contratiempos posteriores.
2. Es recomendable realizar la asignación de los temas de investigación al inicio del semestre, para evitar contratiempos para la realización de los mismos.
3. Es recomendable explicar a los alumnos al inicio de semestre que actividades se realizarán de forma individual y que otras se realizarán de forma grupal, y cuando este sea el caso, definir desde un inicio el número máximo de integrantes por grupo.
4. Es necesario tener un medio de comunicación con los alumnos, adicional al correo electrónico, por lo que se recomienda registrarse en la Universidad Virtual, o crear un grupo para mantener la comunicación con los alumnos y publicar documentos.



## **6. DOCUMENTACIÓN DE APOYO DE “ANÁLISIS Y DISEÑO DE SISTEMAS 1”**

A continuación se presenta el desarrollo de la documentación de apoyo del curso de Análisis y Diseño de Sistemas 1, para lo cual se realizó un libro que contiene las unidades más importantes impartidas a lo largo del curso.

### **6.1 Descripción general**

NOMBRE DE CURSO: Análisis y Diseño de Sistemas 1 (283)

PRE- REQUISITOS: Sistemas de Bases de Datos 1 (774)

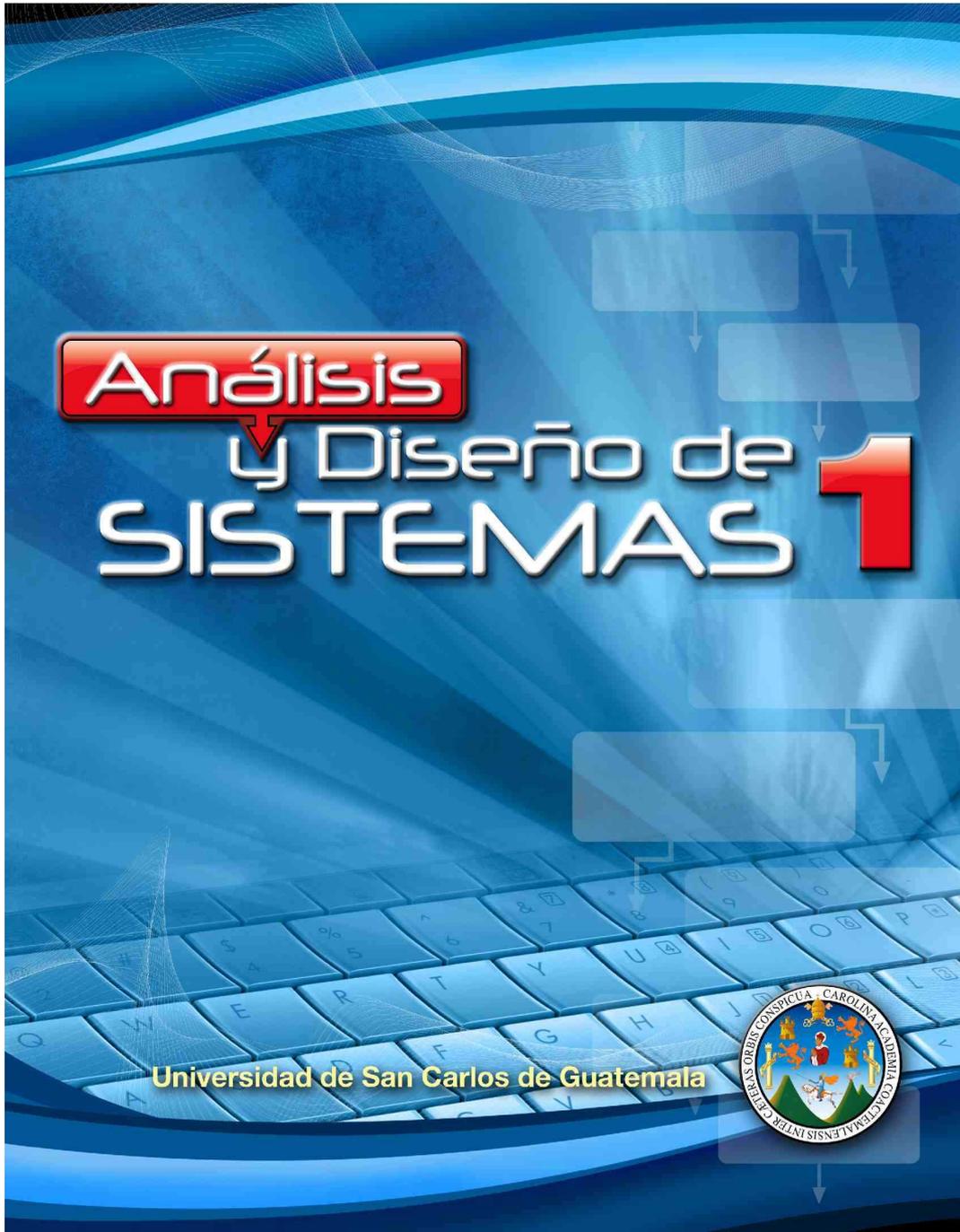
POST – REQUISITOS: Análisis y Diseño de Sistemas 2 (785)

Sistemas Organizacionales y Gerenciales 1 (786)

Sistemas Aplicados 1 (788)

OBJETIVO: Este libro busca apoyar al estudiante de ingeniería de software en los conceptos relacionados con las metodologías de desarrollo, ciclo de vida del desarrollo de software, en el proceso de toma de requerimientos y definición de funcionalidades de los sistemas de información.

Figura 1. Portada Libro Análisis y Diseño de Sistemas 1



## 6.2 Ingeniería de *software*

### Descripción general

Este capítulo presenta en forma general los elementos principales de la ingeniería de *software* y las fases del ciclo de desarrollo de *software*.

### Tabla de contenidos

1. Ingeniería de *software*
  - 1.1. Definición
  - 1.2. Áreas de administración de la ingeniería de *software*
    - 1.2.1. Iniciación y definición del alcance
    - 1.2.2. Planificación del proyecto de *software*
    - 1.2.3. Promulgación del proyecto de *software*
    - 1.2.4. Revisión y evaluación
    - 1.2.5. Cierre
2. Arquitectura de *software*
  - 2.1. Definiciones
3. Ciclo de desarrollo de *software*
  - 3.1. Fases del ciclo de desarrollo de *software*
    - 3.1.1. Investigación preliminar
    - 3.1.2. Análisis del problema
    - 3.1.3. Fase de diseño
    - 3.1.4. Fase de codificación
    - 3.1.5. Fase de pruebas
    - 3.1.6. Fase de implementación y mantenimiento

## 6.3 Métodos de desarrollo de *software*

### Descripción general

En este capítulo se presentan los modelos de desarrollo de *software* clásico, espiral y prototipos. También se incluye una breve introducción a las metodologías de desarrollo ágil, y los aspectos fundamentales de las metodologías de desarrollo RUP y XP, DSDM, *Scrum* y FDD.

### Tabla de contenidos

1. Modelos de desarrollo de *software*
  - 1.1. Modelo de cascada
  - 1.2. Modelo de prototipos
    - 1.2.1. Etapas del desarrollo de prototipos
  - 1.3. Modelo espiral
    - 1.3.1. Actividades en el modelo espiral
  - 1.4. Comparación entre los diferentes modelos
2. Métodos ágiles
  - 2.1. Metodologías ágiles
  - 2.2. Manifiesto ágil
    - 2.2.1. Principios del manifiesto ágil
    - 2.2.2. Valores del manifiesto ágil
3. Modelos
  - 3.1. Modelo CMM
    - 3.1.1. Niveles de CMM
  - 3.2. Modelo CMMI
    - 3.2.1. Modelos de CMMI

- 4. Proceso unificado
  - 4.1. Mejores prácticas de RUP
    - 4.1.1. Desarrollo iterativo
    - 4.1.2. Administración de requerimientos
    - 4.1.3. Utilizar una arquitectura de componentes
    - 4.1.4. Modelado visual
    - 4.1.5. Verificación continua de la calidad
    - 4.1.6. Administración del cambio
  - 4.2. Fases de RUP
    - 4.2.1. Fase de concepción
    - 4.2.2. Fase de elaboración
    - 4.2.3. Fase de construcción
    - 4.2.4. Fase de transición
  - 4.3. Disciplinas de RUP
    - 4.3.1. Modelado del negocio
    - 4.3.2. Requerimientos
    - 4.3.3. Análisis y diseño
    - 4.3.4. Implementación
    - 4.3.5. Pruebas
    - 4.3.6. Despliegue
    - 4.3.7. Administración de la configuración
    - 4.3.8. Administración del proyecto
    - 4.3.9. Ambiente
  - 4.4. Ventajas y desventajas
- 5. *Extreme Programming*
  - 5.1. Prácticas de XP
    - 5.1.1. Escala fina de retroalimentación
    - 5.1.2. Proceso continuo
    - 5.1.3. Compartiendo el conocimiento

- 5.1.4. Bienestar del programador
- 5.2. Principios de XP
- 5.3. Valores de XP
  - 5.3.1. Comunicación
  - 5.3.2. Simplicidad
  - 5.3.3. Retroalimentación
  - 5.3.4. Coraje
  - 5.3.5. Respeto
- 5.4. Fases del ciclo de desarrollo
  - 5.4.1. *Exploration*
  - 5.4.2. *Planning*
  - 5.4.3. *Iterations to release*
  - 5.4.4. *Productionizing*
  - 5.4.5. *Maintenance*
  - 5.4.6. *Death*
- 5.5. Ventajas y desventajas
- 6. DSDM
  - 6.1. RAD
  - 6.2. Fases de DSDM
    - 6.2.1. *Feasibility study*
    - 6.2.2. *Business study*
    - 6.2.3. *Functional model iteration*
    - 6.2.4. *Design and build iteration*
    - 6.2.5. *Implementation*
  - 6.3. Principios
  - 6.4. Técnicas de DSDM
    - 6.4.1. Cajas de tiempo
    - 6.4.2. MoSCoW
    - 6.4.3. Prototipado

- 6.4.4. Pruebas
- 6.4.5. *Workshop*
- 6.4.6. Modelado
- 6.5. Ventajas y desventajas
- 7. *Scrum*
  - 7.1. Fases de *Scrum*
    - 7.1.1. Fase *pre-game*
    - 7.1.2. Fase de desarrollo
    - 7.1.3. Fase de *post-game*
  - 7.2. Roles
    - 7.2.1. Roles "cerdo"
    - 7.2.2. Roles "gallina"
  - 7.3. Prácticas
    - 7.3.1. *Backlog* producto
    - 7.3.2. Estimación de esfuerzo
    - 7.3.3. *Sprint*
    - 7.3.4. Reunión de planificación del *Sprint*
    - 7.3.5. *Sprint Backlog*
    - 7.3.6. Reunión diaria de *Scrum*
    - 7.3.7. Reunión de revisión de *Spring*
  - 7.4. Ventajas y desventajas
- 8. FDD (*Feature Driven Development*)
  - 8.1. Proceso
    - 8.1.1. Desarrollar un modelo general
    - 8.1.2. Construir una lista de características
    - 8.1.3. Planificar por características
    - 8.1.4. Diseñar por características y construir por características
  - 8.2. Roles
    - 8.2.1. Roles clave

- 8.2.2. Roles de soporte
- 8.2.3. Roles adicionales
- 8.3. Prácticas
- 8.4. Ventajas y desventajas

## **6.4 Administración de procesos de negocio**

### **Descripción general**

Este capítulo presenta una breve descripción sobre la administración de procesos de negocio (BPM), incluyendo los aspectos fundamentales de dicha disciplina, entre estos, los principios que lo fundamentan, estándares utilizados en la actualidad, una arquitectura para BPM y el ciclo de vida de los procesos.

### **Tabla de contenidos**

- 1. *Business Process Management*
  - 1.1. Principios
    - 1.1.1. Los procesos son activos
    - 1.1.2. Los procesos deben ser administrados y mejorados continuamente
    - 1.1.3. Las tecnologías de la información son habilitadores esenciales
  - 1.2. Prácticas
    - 1.2.1. Estructura organizacional orientada al proceso
    - 1.2.2. Designar propietarios al proceso
    - 1.2.3. Compromiso descendente, ejecución ascendente
    - 1.2.4. Utilizar tecnologías de la información para administrar los procesos de negocio

- 1.2.5. Colaborar con los socios del negocio
- 1.2.6. Aprendizaje continuo y mejora de procesos
- 1.2.7. Alinear las remuneraciones de empleados a la mejora de procesos
- 1.2.8. Utilizar BPR, TQM y otras herramientas de mejora de proceso
- 2. Ciclo de vida de los procesos
  - 2.1. Análisis y diseño
  - 2.2. Configuración
  - 2.3. Promulgación
  - 2.4. Evaluación
  - 2.5. Administración y *stakeholders*
- 3. Estándares de BPM
  - 3.1. Estándares de definición de proceso
    - 3.1.1. XPDL
    - 3.1.2. BPML
  - 3.2. Estándares de integración de proceso
    - 3.2.1. Wf-XML 2.0
    - 3.2.2. BPEL
    - 3.2.3. WSCI
- 4. Arquitectura BMP
  - 4.1. Arquitectura de sistemas de administración de flujos de trabajo
  - 4.2. *Web services*

## 6.5 Administración de requerimientos con RUP

### Descripción general

El presente capítulo presenta los elementos principales y las características de la disciplina de administración de requerimientos de la metodología de trabajo RUP. Dentro del capítulo se incluyen los principales artefactos de la administración de requerimientos, flujo de trabajo y el modelado de requerimientos mediante casos de uso.

### Tabla de contenidos

1. Administración de requerimientos con RUP
2. Tipos de requerimientos
  - 2.1. Requerimientos funcionales
  - 2.2. Requerimientos no funcionales
    - 2.2.1. ISO 9126
    - 2.2.2. FURPS
3. Flujo de trabajo
  - 3.1. Análisis del problema
  - 3.2. Entender las necesidades de los *stakeholders*
  - 3.3. Definir el sistema
  - 3.4. Administrar el alcance del proyecto
  - 3.5. Refinar la definición del sistema
  - 3.6. Manejar el cambio de requerimientos
4. Artefactos
  - 4.1. Documento de visión
  - 4.2. Especificaciones suplementarias

- 5. Introducción a casos de uso
  - 5.1. Modelo de casos de uso
    - 5.1.1. Actores
    - 5.1.2. Casos de uso
  - 5.2. Flujo de eventos
    - 5.2.1. Flujo básico
    - 5.2.2. Flujos alternos
  - 5.3. Escenarios
  - 5.4. Relaciones entre casos de uso
    - 5.4.1. Inclusión
    - 5.4.2. Extensión
    - 5.4.3. Generalización

## **6.6 Administración de requerimientos con *Extreme Programming***

### **Descripción general**

Este capítulo presenta algunos aspectos relevantes sobre la administración de requerimientos mediante historias de usuario, la forma como lo describe la metodología de desarrollo XP.

## Tabla de contenidos

1. Historias de usuario
  - 1.1. Características de las historias de usuario
    - 1.1.1. Independencia
    - 1.1.2. Negociable
    - 1.1.3. Agrega valor a los usuarios
    - 1.1.4. Estimable
    - 1.1.5. Pequeña
    - 1.1.6. Probable
  - 1.2. Qué no son historias de usuario
    - 1.2.1. Los casos de uso
    - 1.2.2. Los escenarios
  - 1.3. Planificación de historias de usuario
  - 1.4. Partes de una historia de usuario
    - 1.4.1. Ejemplo del esquema de una historia de usuario
    - 1.4.2. Ejemplo de una historia de usuario
2. Estimación
  - 2.1. Estimación de historias de usuario
    - 2.1.1. Puntos de historia
    - 2.1.2. Estimación de historias
  - 2.2. Planificación de *releases*
    - 2.2.1. Priorización de historias

## **6.7 Usabilidad**

### **Descripción general**

En este capítulo presenta los aspectos fundamentales de la usabilidad que deben de tomarse en cuenta para el desarrollo de sistemas de información.

### **Tabla de contenidos**

1. Usabilidad
2. Tipos de atributos de la usabilidad
  - 2.1. Atributos cuantificables de forma objetiva
  - 2.2. Atributos cuantificables de forma subjetiva
3. Áreas de la usabilidad
4. Principios de la usabilidad
  - 4.1. Facilidad de aprendizaje
  - 4.2. Flexibilidad
  - 4.3. Robustez
5. Usabilidad Web
  - 5.1. Conceptos generales
  - 5.2. Elementos principales
6. Accesibilidad
  - 6.1. Psicología del color
  - 6.2. Psicología de algunos colores
  - 6.3. Facilidad de uso
  - 6.4. Estructura
  - 6.5. Navegación
  - 6.6. Gráficos

6.7. Tipografía

7. Evaluar la usabilidad

7.1. Pruebas de usabilidad

## 6.8 Demo: capítulo 1 “Ingeniería de Software”

Figura 2. Páginas 13-16 Libro “Análisis y Diseño de Sistemas 1”

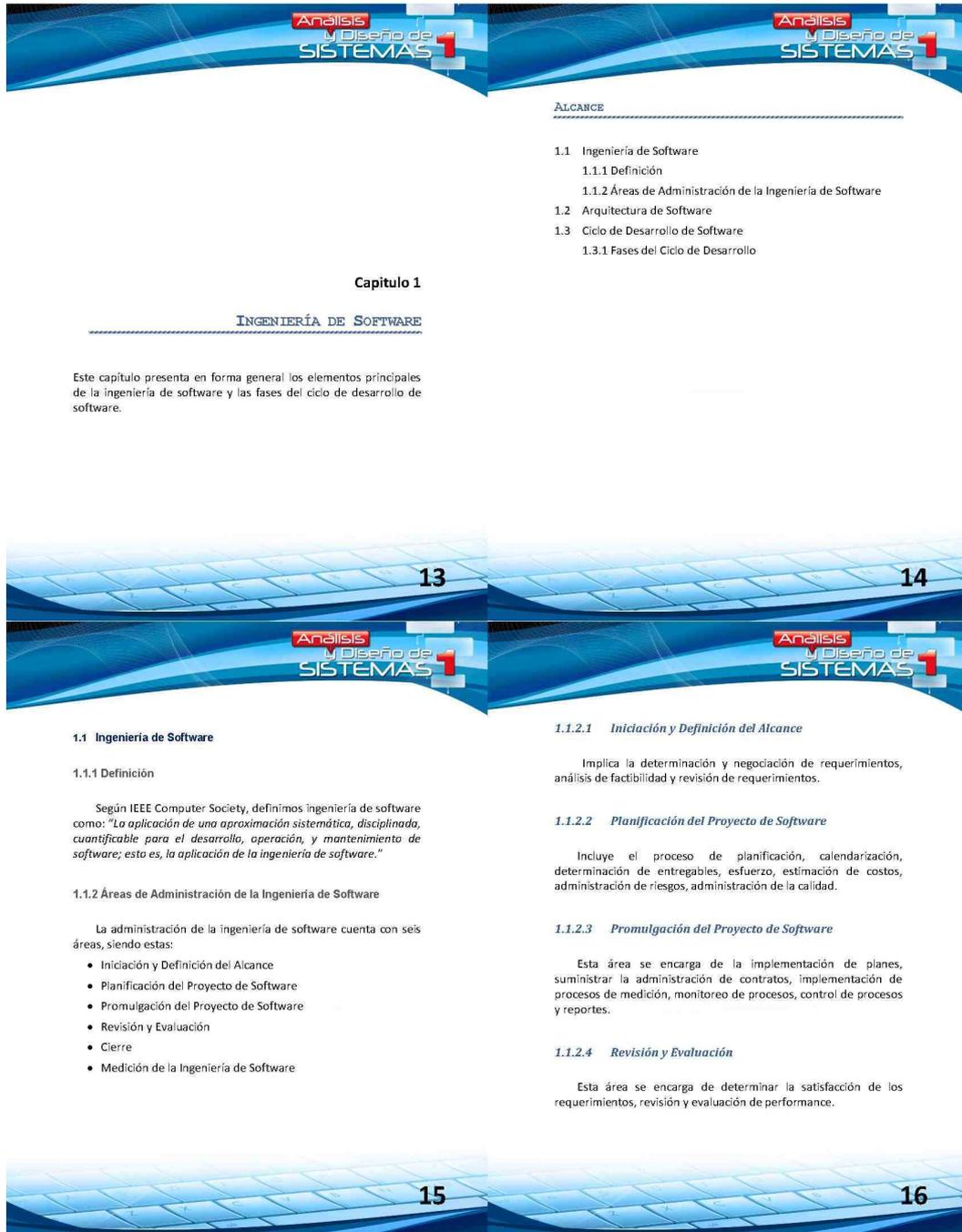


Figura 3. Páginas 17-20 Libro “Análisis y Diseño de Sistemas 1”

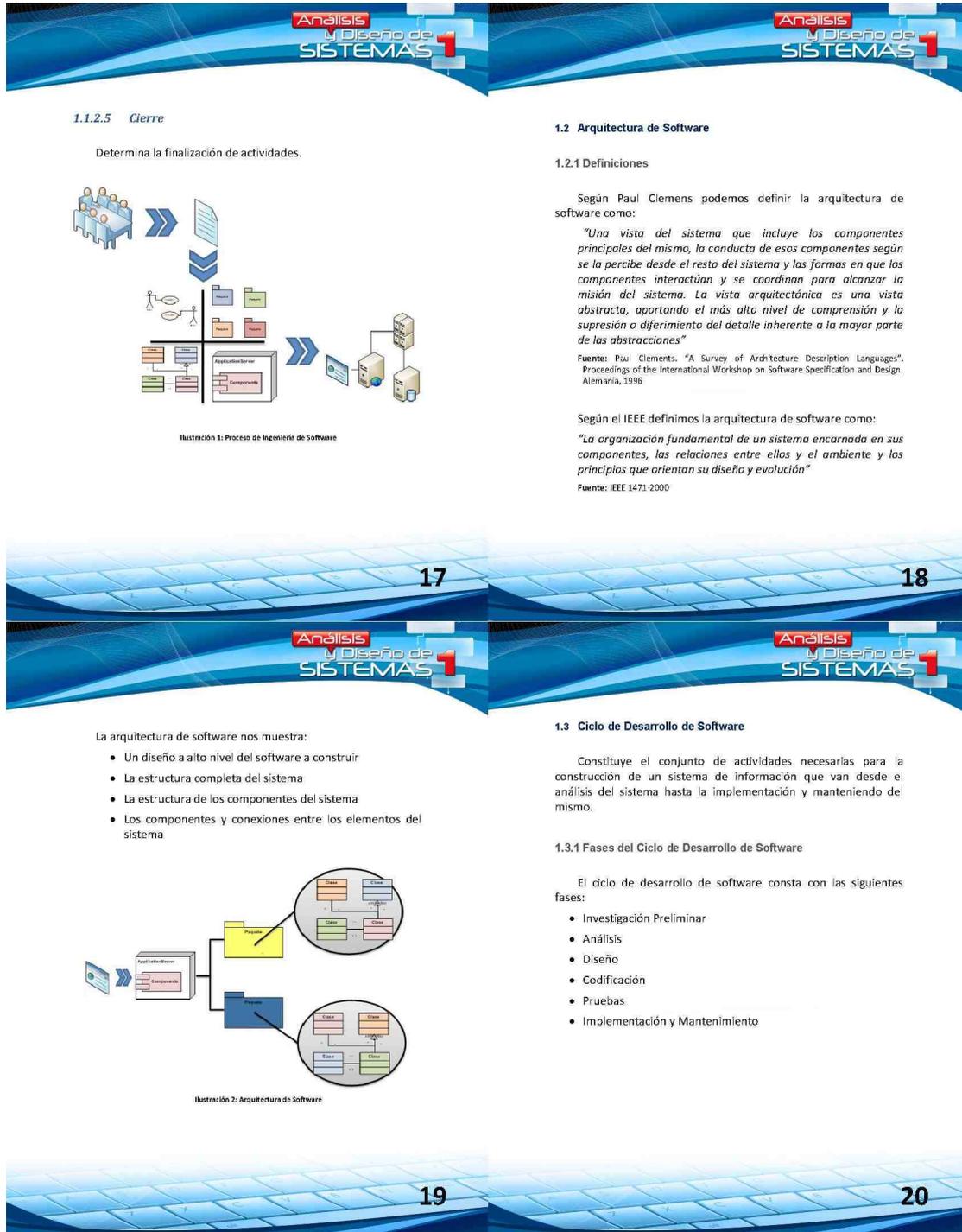


Figura 4. Páginas 21-24 Libro “Análisis y Diseño de Sistemas 1”

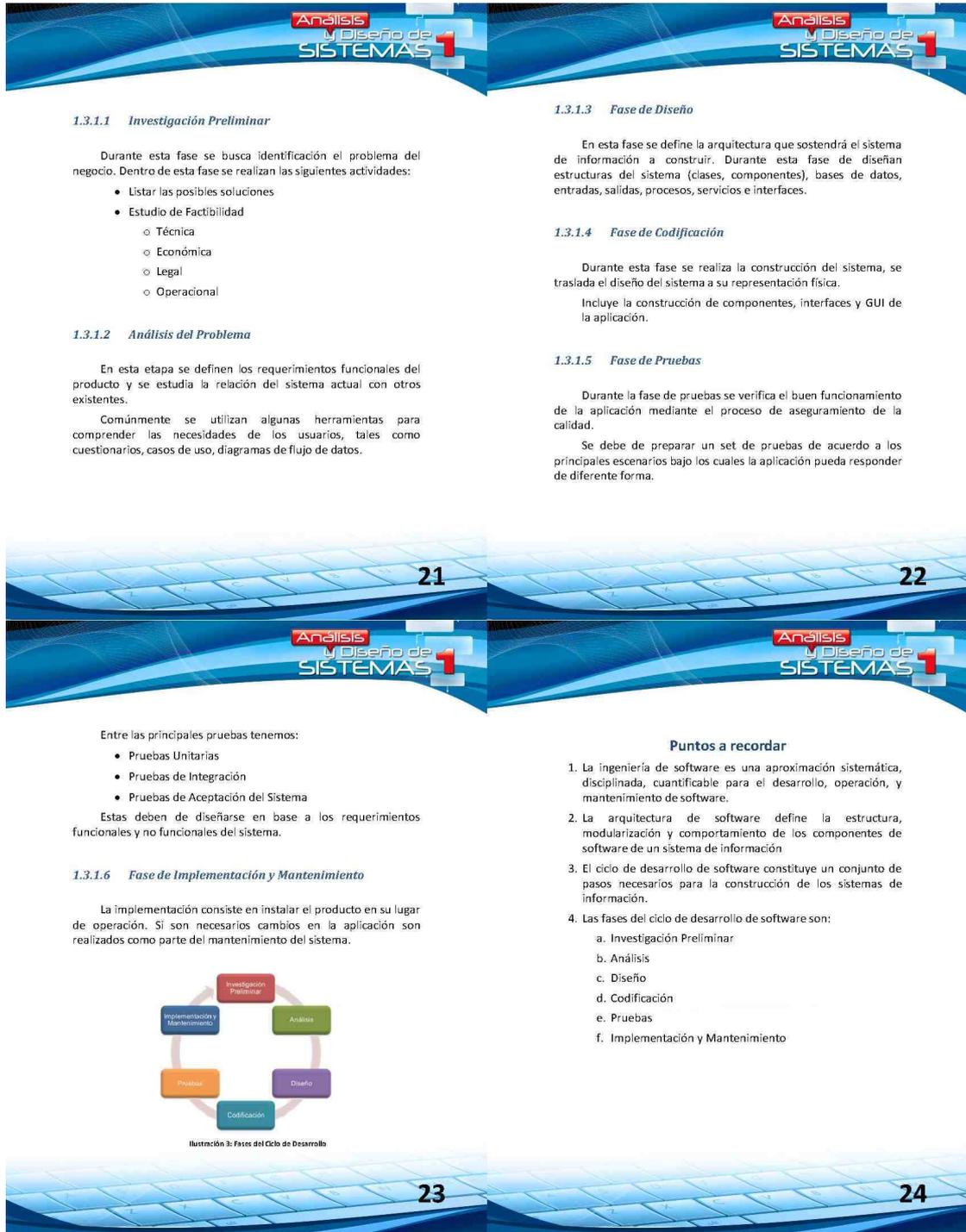
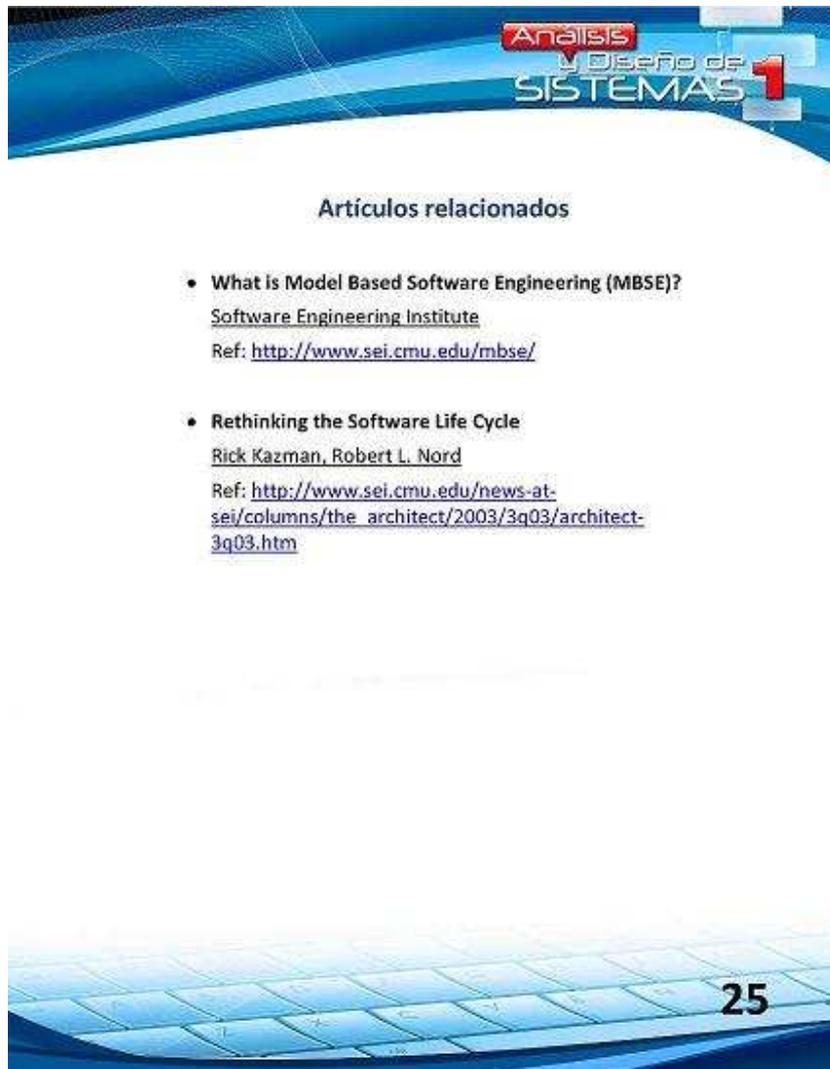


Figura 5. Página 25 Libro “Análisis y Diseño de Sistemas 1”



## 7. DOCUMENTACIÓN DE APOYO DE “ANÁLISIS Y DISEÑO DE SISTEMAS 2”

A continuación se presenta el desarrollo de la documentación de apoyo del curso de Análisis y Diseño de Sistemas 2, para lo cual se realizó un libro que contiene las unidades más importantes impartidas a lo largo del curso.

### 7.1 Descripción general

NOMBRE DE CURSO: Análisis y Diseño de Sistemas 2 (785)

PRE- REQUISITOS: Análisis y Diseño de Sistemas 1 (283)

POST – REQUISITOS: *Software Avanzado* (780)  
*Sistemas Aplicados 2* (789)  
*Auditoría de Proyectos de Software* (735)

OBJETIVO: Este libro busca apoyar al estudiante de ingeniería de *software* en los conceptos de administración de la configuración, integración continua, definición de arquitecturas de *software* y SOA.

Figura 6. Portada del Libro de Análisis y Diseño de Sistemas 2



## 7.2 Administración de la configuración

### Descripción general

En este capítulo se definen los elementos de *software* sobre los cuales debe de llevarse un control de cambios. También se define un flujo de trabajo ordenado para el control de cambios y algunas herramientas utilizadas para realizar dichas tareas.

### Tabla de contenidos

1. Administración de la configuración
2. Conceptos generales
  - 2.1. Repositorio
    - 2.1.1. Modelos de manejo de repositorio
    - 2.1.2. Operaciones en repositorios
  - 2.2. Espacio de trabajo
  - 2.3. Modelos de versionamiento
    - 2.3.1. Bloquear-modificar-desbloquear
    - 2.3.2. Copiar-modificar-fusionar
  - 2.4. Árbol, rama y etiqueta
    - 2.4.1. Árbol de versiones
    - 2.4.2. Rama
    - 2.4.3. Etiqueta
3. Nomenclatura de versiones
  - 3.1. Identificadores basados en secuencia
  - 3.2. Fecha
  - 3.3. Año de *release*
  - 3.4. Códigos alfanuméricos

4. Administración del cambio
  - 4.1. Administración de las peticiones de cambio
  - 4.2. Configuración de la contabilidad de estados
  - 4.3. Administración de la configuración
  - 4.4. Seguimiento del cambio
  - 4.5. Selección de versión
  - 4.6. Manufacturación del *software*
5. Herramientas para la administración de la configuración
  - 5.1. Subversion
  - 5.2. Maven
  - 5.3. Microsoft *Source Safe*

## **7.3 Integración continua**

### **Descripción general**

En este capítulo se presentan las características principales de la integración continua, las prácticas que la hacen más eficiente, el proceso de integración continua y los beneficios que la misma ofrece para el desarrollo de software.

## Tabla de contenidos

1. Integración continua
2. Características de la integración continua
  - 2.1. Compilación de código fuente
  - 2.2. Integración de base de datos
  - 2.3. Pruebas
  - 2.4. Inspección
  - 2.5. Despliegue
  - 2.6. Documentación y retroalimentación
3. Beneficios
  - 3.1. Reducción de riesgos
  - 3.2. Reducir el proceso repetitivo manual
  - 3.3. Generar *software* desplegable en cualquier momento y lugar
  - 3.4. Habilitar una mejor visibilidad del proyecto
  - 3.5. Establecer una gran confianza en el producto
4. Prácticas
  - 4.1. Hacer *commit* del código frecuentemente
  - 4.2. No hacer *commit* código roto
  - 4.3. Reparar *builds* rotos de forma inmediata
  - 4.4. Escribir pruebas de desarrollo automatizadas
  - 4.5. Todas las pruebas e inspecciones deben de pasar
  - 4.6. Correr *builds* privados
  - 4.7. Evitar tener código roto
5. Proceso continuo
6. Complemento con otras prácticas de desarrollo
7. Reduciendo el riesgo
  - 7.1. Pérdida de *software* desplegable
  - 7.2. Descubrimiento tardío de defectos

- 7.3. Pérdida de visibilidad del proyecto
- 7.4. *Software* de poca calidad
- 8. Construyendo *software* en cada cambio
- 9. Herramientas de integración continua
  - 9.1. Apache Continuum
  - 9.2. CruiseControl
  - 9.3. Bamboo

## **7.4 Arquitectura de *software***

### **Descripción general**

En este capítulo se presentan algunos elementos de la arquitectura de *software*, entre estos: atributos de calidad, *middleware*, arquitectura orientada aspectos y arquitectura dirigida por modelos. También se presenta una vista general del proceso para definición de la arquitectura de un sistema, y algunos instrumentos útiles para documentar la arquitectura de un sistema de información.

### **Tabla de contenidos**

- 1. Arquitectura de *software*
- 2. Atributos de calidad del *software*
  - 2.1. Rendimiento
  - 2.2. Escalabilidad
  - 2.3. Seguridad
  - 2.4. Disponibilidad
  - 2.5. Portabilidad

- 2.6. Soporte
- 3. Arquitectura *middleware*
  - 3.1. *Middleware* orientado a mensajes
    - 3.1.1. Publicador – suscriptor
  - 3.2. Servidor de aplicaciones
  - 3.3. *Message Broker*
  - 3.4. Orquestador de procesos de negocio
- 4. Proceso para definición arquitectura de software
  - 4.1. Determinar los requerimientos de la arquitectura
  - 4.2. Identificar los requerimientos de la arquitectura
  - 4.3. Priorizar los requerimientos de la arquitectura
  - 4.4. Diseñar la arquitectura
    - 4.4.1. Seleccionar un *framework* de arquitectura
    - 4.4.2. Asignar componentes
  - 4.5. Validación
- 5. Documentación de arquitectura
  - 5.1. Modelo 4+1 vistas
    - 5.1.1. Vista de casos de uso
    - 5.1.2. Vista lógica
    - 5.1.3. Vista de implementación
    - 5.1.4. Vista de procesos
    - 5.1.5. Vista de despliegue
  - 5.2. Arquitectura Bredemeyer
    - 5.2.1. Meta arquitectura
    - 5.2.2. Arquitectura
    - 5.2.3. Guías de arquitectura y políticas
- 6. Arquitectura orientada a aspectos
  - 6.1. Conceptos
  - 6.2. Relación con la arquitectura de *software*

- 7. Arquitectura dirigida por modelos (MDA)
  - 7.1. Modelos MDA
    - 7.1.1. *Computation independent model* (CIM)
    - 7.1.2. *Platform independent model*
    - 7.1.3. *Platform specific model* (PSM)
    - 7.1.4. *Implementation specific model*
  - 7.2. Transformaciones en MDA
    - 7.2.1. Refinamientos
    - 7.2.2. *Refactoring*
    - 7.2.3. Puentes
    - 7.2.4. Anti-refinamientos
  - 7.3. Beneficios
    - 7.3.1. Portabilidad
    - 7.3.2. Interoperabilidad
    - 7.3.3. Reusabilidad
  - 7.4. Relación con la arquitectura de *software*

## **7.5 Arquitectura orientada a servicios**

### **Descripción general**

En este capítulo se presentan los fundamentos de la arquitectura orientada a servicios, los elementos y estándares de la comunicación e integración mediante servicios Web, principios elementales de la orientación a servicios y las capas que conforman una arquitectura de servicios.

### **Tabla de contenidos**

1. Arquitectura orientada a servicios
2. Beneficios
  - 2.1. Mejora la integración
  - 2.2. Reutilización
  - 2.3. Arquitecturas y soluciones modernizadas
  - 2.4. Establece una representación de datos XML estandarizada
  - 2.5. Se enfoca en la inversión en la infraestructura de comunicación
  - 2.6. Flexibilidad en las aplicaciones
  - 2.7. Manejabilidad de las aplicaciones
3. Fundamentos de SOA
  - 3.1. Analogía de orientación a servicios
  - 3.2. Encapsular la lógica de negocio
  - 3.3. Relación entre servicios
  - 3.4. Comunicación entre servicios
  - 3.5. Diseño de servicios
4. Evolución de SOA
5. Servicios Web

- 5.1. *Framework*
- 5.2. Roles
- 5.3. Modelos
- 5.4. SOAP
- 5.5. WSDL y UDDI
- 5.6. UDDI
- 6. Principios de orientación al servicio
  - 6.1. Reutilización de los servicios
  - 6.2. Compartir un contrato formal
  - 6.3. Débilmente acoplados
  - 6.4. Abstracción de la lógica
  - 6.5. Composición de servicios
  - 6.6. Autonomía
  - 6.7. Los servicios son sin estado
  - 6.8. Descubrimiento
- 7. Anatomía de SOA
  - 7.1. Componentes lógicos del marco de servicios Web
  - 7.2. Componentes lógicos de automatización de la lógica
  - 7.3. Componentes de SOA
  - 7.4. ¿Cómo se interrelacionan los componentes en SOA?
- 8. Capas de servicios
  - 8.1. Capa de servicios de aplicación
  - 8.2. Capa de servicios de negocio
  - 8.3. Capa de servicios de orquestación

## **7.6 Análisis de clases**

### **Descripción general**

Este capítulo muestra cómo realizar el análisis de los requerimientos de un sistema y transformarlos en una representación de diseño, definiendo las principales clases que conformarán parte de la arquitectura del sistema. Esto se realizará por medio de la herramienta de análisis de clases soportada por RUP.

También se presenta una breve introducción al patrón de diseño MVC.

### **Tabla de contenidos**

1. Patrón modelo-vista-controlador
  - 1.1. Principio abierto-cerrado (OCP)
  - 1.2. Elementos de MVC
    - 1.2.1. El modelo
    - 1.2.2. La vista
    - 1.2.3. El controlador
  - 1.3. Flujo básico
  - 1.4. Ventajas y desventajas
2. Análisis de casos de uso
  - 2.1. Realización de casos de uso
  - 2.2. Descripción suplementaria de casos de uso
  - 2.3. Buscar clases de análisis a partir del comportamiento de los casos de uso
  - 2.4. Distribuir el comportamiento a las clases de análisis
  - 2.5. Describir responsabilidades

- 2.6. Describir atributos y asociaciones
- 3. Encontrando clases a partir de casos de uso
  - 3.1. Clases límite
  - 3.2. Clases de entidad
  - 3.3. Clases de control
  - 3.4. Representación gráfica de las clases de análisis
  - 3.5. Ejemplo: traslado de fondos
- 4. Modelo de análisis

## 7.7 Patrones de diseño

### Descripción general

Este capítulo se presenta una breve introducción al tema “patrones de diseño”, los cuales proponen estructuras de comportamiento para diferentes escenarios de funcionalidad de sistemas. Dentro del capítulo se desarrollan tres tipos de patrones: patrones de creación, de estructura y de comportamiento.

### Tabla de contenidos

- 1. Patrones de diseño
- 2. Patrones de creación
  - 2.1. *Factory Method* (método de fabricación)
    - 2.1.1. Cuándo usar este patrón
    - 2.1.2. Elementos del patrón
    - 2.1.3. Ejemplo
  - 2.2. *Abstract Factory* (fábrica abstracta)
    - 2.2.1. Cuándo usar este patrón

- 2.2.2. Elementos del patrón
    - 2.2.3. Ejemplo
  - 2.3. *Singleton* (único)
    - 2.3.1. Cuándo usar este patrón
    - 2.3.2. Elementos del patrón
    - 2.3.3. Ejemplo
- 3. Patrones de estructura
  - 3.1. *Adapter* (adaptador)
    - 3.1.1. Cuándo usar este patrón
    - 3.1.2. Elementos del patrón
    - 3.1.3. Ejemplo
  - 3.2. *Composite* (compuesto)
    - 3.2.1. Cuándo usar este patrón
    - 3.2.2. Elementos del patrón
    - 3.2.3. Ejemplo
  - 3.3. *Decorator* (decorador)
    - 3.3.1. Cuándo usar este patrón
    - 3.3.2. Elementos del patrón
    - 3.3.3. Ejemplo
  - 3.4. *Facade* (fachada)
    - 3.4.1. Cuándo usar este patrón
    - 3.4.2. Elementos del patrón
    - 3.4.3. Ejemplo
  - 3.5. *Proxy* (apoderado)
    - 3.5.1. Cuándo usar este patrón
    - 3.5.2. Elementos del patrón
    - 3.5.3. Ejemplo
- 4. Patrones de comportamiento
  - 4.1. *Iterator* (iterador)

- 4.1.1. Cuándo usar este patrón
  - 4.1.2. Elementos del patrón
  - 4.1.3. Ejemplo
- 4.2. *Observer* (observador)
  - 4.2.1. Cuándo usar este patrón
  - 4.2.2. Elementos del patrón
  - 4.2.3. Ejemplo
- 5. *Command* (orden)
  - 5.1.1. Cuándo usar este patrón
  - 5.1.2. Elementos del patrón
  - 5.1.3. Ejemplo
- 6. *State* (estado)
  - 6.1.1. Cuándo usar este patrón
  - 6.1.2. Elementos del patrón
  - 6.1.3. Ejemplo
- 7. *Strategy* (estrategia)
  - 7.1.1. Cuándo usar este patrón
  - 7.1.2. Elementos del patrón
  - 7.1.3. Ejemplo
- 8. *Template Method* (método plantilla)
  - 8.1.1. Cuándo usar este patrón
  - 8.1.2. Elementos del patrón
  - 8.1.3. Ejemplo

## 7.8 Demo: capítulo 1 “Administración de la Configuración”

Figura 7. Páginas 18-21 Libro “Análisis y Diseño de Sistemas 2”

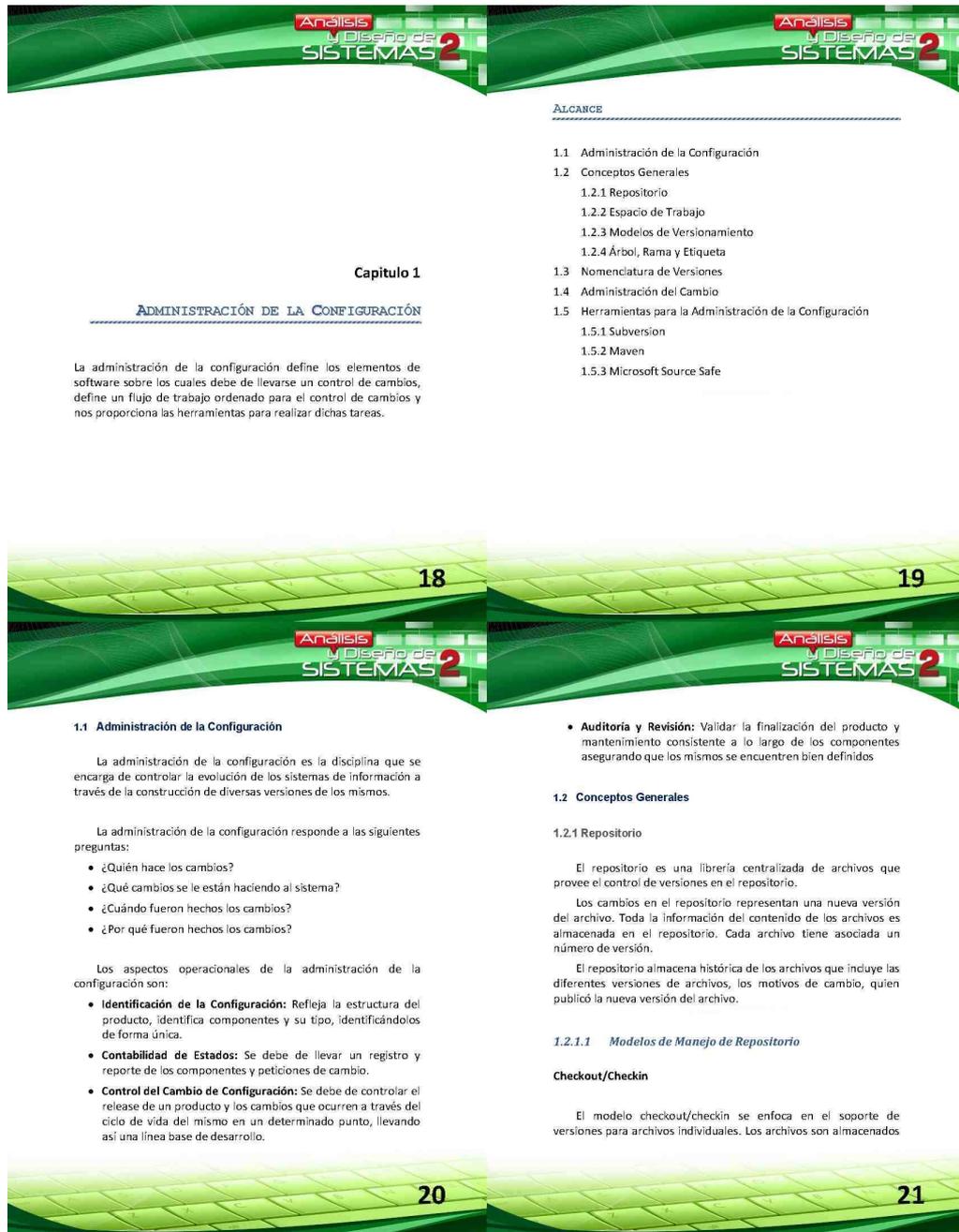


Figura 8. Páginas 22-25 Libro “Análisis y Diseño de Sistemas 2”



y versionados en el repositorio. La creación de nuevas versiones es controlada en el repositorio.

Los archivos no pueden ser accedidos directamente al repositorio, sino que son accedidos a través de las operaciones de checkout (tomar una versión existente del archivo para lectura) y checkin (crear una nueva versión del archivo en el repositorio).

El repositorio debe de llevar control para acceso concurrente de los archivos mediante las operaciones de checkin y checkout.

Las características principales del modelo son el mantenimiento el historial de versiones de archivos individuales y su control para modificaciones concurrentes.

En este modelo el usuario trabajo con el repositorio y con el sistema de archivos. Los archivos son versionados y almacenados en el sistema de archivos, mientras que la creación de una nueva versión es manejada por la herramienta del repositorio.

El modelo checkout/checkin provee los siguientes conceptos para versionado de archivos:

- Evolución de una versión secuencial histórica
- Creación de ramas
- Unión de dos versiones de diferentes ramas en una nueva versión en una de las ramas

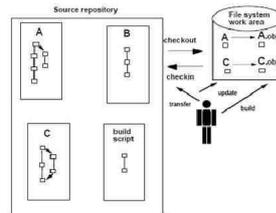


Ilustración 1: Modelo Checkout/Checkin  
Fuente: Configuration Management Models  
In Commercial Environments, Peter H. Feller

#### Composición

Se enfoca en los conceptos de repositorio, área de trabajo y el manejo concurrente a través del bloqueo de componentes mientras es construido en las propiedades de una versión del componente. Busca mejorar el soporte para crear configuraciones, para su manejo histórico y para su uso como contextos de trabajo sobre un sistema de administración de configuración. Este modelo las configuraciones son entidades entendibles por el sistema de administración de la configuración.

Los desarrolladores operan con configuraciones de forma repetitiva formando un sistema a partir de sus componentes y seleccionando la versión deseada para cada componente.

- El modelo de composición utiliza la configuración de un modelo del sistema y reglas de selección de versión.
- El modelo del sistema lista todos los componentes que conforman el sistema.

Las reglas de sección de versión indican cual versión será escogida para cada componente para construir una configuración.

Las reglas de selección son aplicadas al modelo del sistema, seleccionando la versión de un componente.

El modo de operación de este modelo consiste en que un desarrollador define el sistema en términos de sus componentes y en un paso separado selecciona una versión apropiada para cada componente.

La selección puede ocurrir en cualquier nivel de composición del sistema, y mediante un sistema para administración de la configuración proporcionar flexibilidad para administrar las variantes del sistema.

Los componentes que conforman el modelo del sistema pueden ser agrupados y los agregados pueden formar parte de otro grupo, construyendo así una estructura jerárquica, la cual puede ser enriquecida con dependencias.

La información representada en los modelos de sistema puede estar contenida en muchos lugares, por lo que su representación en una estructura de directorios ayuda a mejorar su organización.

La administración de la configuración nos ayuda a mantener actualizadas las fuentes de un conjunto de componentes.

Las reglas de selección hacen única cada versión de un componente. Las reglas de selección se pueden clasificar en dos tipos:

- Uno es cuando son vistas como caminos de búsqueda en un grafo de versiones con versiones etiquetadas y ramas. En un camino de búsqueda el usuario indica la secuencia de opciones de selección conforme las versiones del componente son seleccionadas.
- El otro es basado en objetos y atributos. Los componentes y sus versiones son vistas como objetos con atributos. Los atributos incluyen el identificador de la versión, fecha de modificación, estado del objeto, entre otros.

Figura 9. Páginas 26-29 Libro “Análisis y Diseño de Sistemas 2”

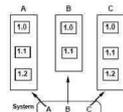


Ilustración 2: Modelo de Composición  
Fuente: Configuration Management Models  
In Commercial Environments, Peter H. Feller

**Transacción Larga**

Este modelo se enfoca en administrar la evolución del sistema completo como series de cambios atómicos y coordinar los cambios del sistema a través de equipos y desarrolladores.

Los desarrolladores trabajan con las configuraciones. Los cambios son realizados en transacciones y una configuración en particular es seleccionada como el punto de partida para los cambios.

Cuando se realizan múltiples transacciones son coordinadas a través esquemas de control de concurrencia que garanticen que no existan pérdidas de cambios.

El resultado de una transacción cometida (commit) es una nueva versión de la configuración. El resultado de una serie de

cambios secuenciales es una secuencia de versiones que constituyen un camino de desarrollo.

Las versiones pueden salir de una rama del camino de desarrollo, resultando en un nuevo e independiente camino de desarrollo.

En este modelo, los desarrolladores trabajan de primero con versiones de configuración. Primeramente seleccionan la versión de la configuración del sistema, después se enfocan en la estructura. La versión de los componentes puede ser determinada de forma implícita por configuración.

El modelo consta de dos conceptos:

- Espacio de Trabajo
- Esquema de control de Concurrencia.

El espacio de trabajo consiste en una configuración de trabajo y una serie de configuraciones preservadas en las cuales los componentes y la estructura del sistema puede ser modificado. El espacio de trabajo es extraído del repositorio y todos los cambios se realizan en el espacio de trabajo. La historia de cambios se realiza de forma local. En el espacio de trabajo existen puntos de chequeo en los cuales es posible revertir el trabajo realizado en el espacio de trabajo. Los cambios no son visibles externamente hasta que son cometidos en el repositorio.



26



27

La administración concurrente puede dividirse en tres categorías diferentes:

- Concurrencia con un espacio de trabajo.
- Concurrencia entre espacios de trabajo diferentes utilizando coordinación.
- Desarrollo concurrente independiente.

Las primeras dos categorías realizan cambios concurrentes a un sistema de configuración, mientras que el tercero asume que un sistema evoluciona en caminos de desarrollo independientes.

La concurrencia con un espacio de trabajo ocurre cuando diversos desarrolladores trabajan en espacio de trabajo separados y el conjunto de sus cambios conforman el sistema. Para este caso existen dos esquemas de control de concurrencia, el conservativo y el optimista. El conservativo requiere un bloqueo a priori a través de los espacios de trabajo. El optimista permite que las modificaciones ocurran concurrentemente y los conflictos son detectados al momento de realizar commit.

La coordinación entre espacios de trabajo ocurre cuando diversos desarrolladores trabajan en espacio de trabajo separados y el conjunto de sus cambios conforman el sistema. Para este caso existen dos esquemas de control de concurrencia, el conservativo y el optimista. El conservativo requiere un bloqueo a priori a través de los espacios de trabajo. El optimista permite que las modificaciones ocurran concurrentemente y los conflictos son detectados al momento de realizar commit.

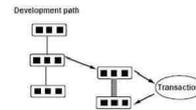


Ilustración 3: Modelo de Transacción Larga  
Fuente: Configuration Management Models  
In Commercial Environments, Peter H. Feller

**Conjunto de Cambios**

El modelo de conjunto de cambios se enfoca en la administración de los cambios lógicos a la configuración del sistema. El concepto de conjunto de cambios representa un conjunto de modificaciones a diferentes componentes para conformar un cambio lógico. Los usuarios de un sistema basado en un conjunto de cambios pueden directamente operar con los conjuntos de cambios.

Las configuraciones pueden ser descritas como una línea base y un conjunto de conjuntos de cambios. Los cambios son propagados a otras configuraciones incluyendo un conjunto de cambios.

Los desarrolladores pueden dar seguimiento a los cambios y determinar si forman parte de una configuración en específico.



28



29

Figura 10. Páginas 30-33 Libro “Análisis y Diseño de Sistemas 2”

**Análisis y Diseño de SISTEMAS 2**

Este tipo de administración de la configuración es conocido como administración de la configuración orientada al cambio ya que se enfoca en cambios lógicos.

Este modelo implica peticiones de cambio, las cuales son herramientas para controlar la inicialización, evaluación, autorización y aprobación de cambios. Debe de existir un registro del estado de un cambio en el proceso de software. Mientras las peticiones de cambio contienen información acerca del cambio, los conjuntos de cambios representan las modificaciones actuales.

Un conjunto de cambios es un conjunto de diferencias entre dos versiones de un archivo. Este conjunto de diferencias entre componentes que deben de ser modificadas entre dos versiones distintas.

Los desarrolladores nombran las versiones de componentes y las configuraciones.

Este modelo ve la configuración como una línea base de configuración más una combinación de conjuntos de cambios.

El modelo permite la creación de nuevas configuraciones agregando una combinación de cambios a una línea base de configuración. Algunos conjuntos de cambios son restringidos por la precedencia que tengan con otros conjuntos de cambios, mientras que otros pueden entrar en conflicto entre sí. Estos conflictos pueden ocurrir cuando diversos programadores trabajen sobre distintas configuraciones de un mismo archivo.

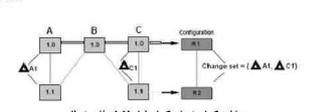


Ilustración 4: Modelo de Conjunto de Cambios  
Fuente: Configuration Management Models  
In Commercial Environments, Peter H. Feller

**1.2.1.2 Operaciones en Repositorios**

**Check in**

Ocurre cuando un cambio local es escrito en el repositorio.

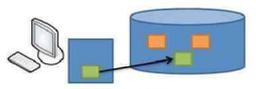


Ilustración 5: Check in

**Check Out**

Creación de una copia local de una porción del repositorio.

---

**Análisis y Diseño de SISTEMAS 2**

30

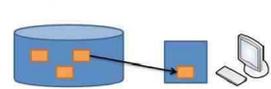


Ilustración 6: Check out

**Importar**

Realiza una copia de un árbol de directorios locales al repositorio por primera vez.

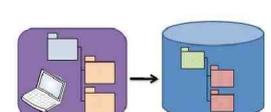


Ilustración 7: Importar

**Exportar**

A diferencia del check out, crea un árbol de directorios en blanco.

**Análisis y Diseño de SISTEMAS 2**

31

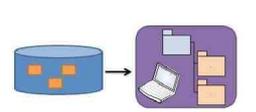


Ilustración 8: Exportar

**Merge**

La operación merge combina las modificaciones a un archivo que han ocurrido de forma independiente en dos ramas versionadas uniéndola en una nueva versión en una de las ramas.

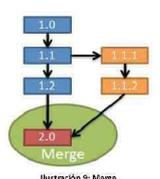


Ilustración 9: Merge

---

**Análisis y Diseño de SISTEMAS 2**

32

**Análisis y Diseño de SISTEMAS 2**

33

Figura 11. Páginas 34-37 Libro “Análisis y Diseño de Sistemas 2”



**Update**

Consiste en sincronizar la copia local con el repositorio. La sincronización puede ser obteniendo las versiones del repositorio y sobrescribiendo los archivos locales, obteniendo las versiones del repositorio y manteniendo los archivos locales modificados.

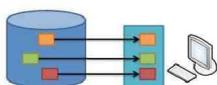


Ilustración 10: Update

**Commit**

Esta operación permite actualizar el repositorio incorporando una nueva versión de un archivo a partir de una copia local (en el workspace).

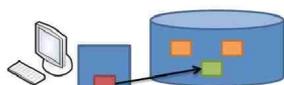


Ilustración 11: Operación Commit

**1.2.2 Espacio de Trabajo**

Es una copia local del repositorio (o de una porción de él) que está sincronizada con el repositorio.

El espacio de trabajo permite realizar cambios de forma local, llevar un registro local de dichos cambios, llevar un control de cambios separado del repositorio, estabilidad del espacio de trabajo bajo el control del desarrollador y control de accesos.

El espacio de trabajo procede de una configuración existente en el repositorio.

El espacio de trabajo consta de dos elementos:

- Configuración de trabajo.
- Series de configuraciones preservadas.

La configuración de trabajo representa la configuración en la cual los componentes y la estructura del sistema pueden ser modificados. Al inicio se utiliza la configuración original, y se va modificando con el transcurrir del tiempo. Las versiones modificadas de los componentes y la estructura del sistema son realizadas de forma local en el espacio de trabajo.

34

35

Se lleva un registro histórico de cambios para ser preservada en una secuencia de configuraciones. Para preservar una configuración es necesario crear una versión de todos los componentes modificados.

La secuencia de configuraciones preservadas provee puntos de chequeo en los cuales los desarrolladores pueden revertir cambios.

Para que los cambios puedan ser reflejados fuera del espacio de trabajo, debe de realizarse una operación de commit a la configuración del repositorio. Después de que la configuración es actualizada en el repositorio, el espacio de trabajo debe de ser eliminado.

Mientras los desarrolladores trabajan en un espacio de trabajo se encuentran aislados de los cambios que se realizan en otros espacios de trabajo y de los cambios realizados en el repositorio.

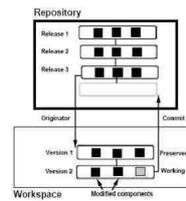


Ilustración 12: Espacio de Trabajo

Fuente: Configuration Management Models In Commercial Environments, Peter H. Feller

**1.2.3 Modelos de Versionamiento**

Cuando existen dos desarrolladores modificando el mismo archivo, y el primer desarrollador graba sus cambios en el repositorio, es posible que cuando el segundo trate de grabar sus cambios se sobrescriban los cambios del primer desarrollador y la nueva versión no cuente con los cambios del primer desarrollador.

36

37

Figura 12. Páginas 38-41 Libro “Análisis y Diseño de Sistemas 2”

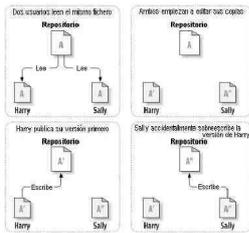


Ilustración 13: Problema al Compartir Archivos  
Fuente: tortoisesvn.net

Para solventar este problema se han desarrollado dos modelos de versionado, siendo estos:

- Bloquear-Modificar-Desbloquear
- Copiar-Modificar-Fusionar

### 1.2.3.1 Bloquear-Modificar-Desbloquear

El repositorio sólo permite que una persona cambie un fichero al mismo tiempo.

El primer desarrollador debe bloquear el archivo antes de que pueda empezar a hacer cambios en él y evitando que el segundo desarrollador pueda hacer cambios en el archivo. El segundo desarrollador únicamente puede leer el archivo y esperar a que el primer desarrollador termine sus cambios y libere su bloqueo. Después de que el archivo haya sido desbloqueado el segundo desarrollador puede bloquear y editar el archivo.

El problema con el modelo bloquear-modificar-desbloquear es que es un poco restrictivo, y a menudo se convierte en un cuello de botella para los usuarios:

- El bloqueo causa muchos problemas administrativos. Por ejemplo si el recurso no es liberado por el primer usuario, el segundo usuario debe de esperar hasta que este sea liberado, o contactar al administrador para solicitar el desbloqueo del recurso.
- El bloqueo puede causar procesos en serie innecesarios. Por ejemplo si el segundo desarrollador únicamente desea agregar código al final del fichero y no afectará los cambios del primer desarrollador, debería de poder efectuar sus cambios sin necesidad de esperar a que el recurso se libere.
- El bloqueo puede causar una falsa sensación de seguridad. Si el primer desarrollador bloquea y edita el fichero A, mientras

que el segundo simultáneamente bloquea y edita el fichero B, si A y B dependen uno del otro, y los cambios hechos a cada uno son semánticamente incompatibles. A y B ya no funcionan juntos. El sistema de bloqueo no tiene forma de prevenir este problema, sin embargo, de alguna forma da una sensación de falsa seguridad.

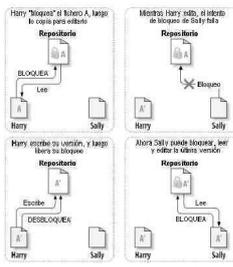


Ilustración 14: Solución Bloquear-Modificar-Desbloquear  
Fuente: tortoisesvn.net

### 1.2.3.2 Copiar-Modificar-Fusionar

En este modelo, el cliente de cada usuario lee el repositorio y crea una copia de trabajo personal del fichero o del proyecto. Luego, los usuarios trabajan en paralelo, modificando sus copias privadas. Finalmente, las copias privadas se fusionan juntas en una nueva versión final. El sistema de control de versiones a menudo ofrece ayuda en la fusión, pero al final la persona es la responsable de hacer que ocurra correctamente.

Si los cambios de un desarrollador se superponen a los de otro, la situación se denomina un conflicto. Cuando un desarrollador le pide a su cliente que fusione los últimos cambios del repositorio en su copia de trabajo, su copia del fichero A se marca de alguna forma como que está en un estado de conflicto: él será capaz de ver ambos conjuntos de cambios conflictivos, y manualmente podrá elegir entre ellos. El software no puede resolver conflictos automáticamente; sólo los humanos son capaces de entender y hacer las elecciones necesarias de forma inteligente. Una vez que el usuario haya resuelto manualmente los cambios que se superponían, puede volcar de forma segura el fichero fusionado al repositorio.

En la práctica, el modelo funciona muy bien. Los usuarios pueden trabajar en paralelo, sin que tengan que esperar nunca uno por otro. Cuando trabajan en los mismos ficheros, resulta que la

Figura 13. Páginas 42-45 Libro “Análisis y Diseño de Sistemas 2”



mayoría de los cambios concurrentes no se superponen en absoluto; los conflictos no son frecuentes. Y el tiempo que lleva resolver conflictos es mucho menor que el tiempo perdido por un sistema bloqueante.

Al final, todo se reduce a un factor crítico: la comunicación entre usuarios. Cuando los usuarios se comunican de forma pobre, aumentan los conflictos sintácticos y semánticos. No hay sistema capaz de forzar a los usuarios a comunicarse perfectamente, y no hay sistema que pueda detectar conflictos semánticos. Por lo que no hay motivo para que se le prometa falsamente que un sistema con bloqueos prevendrá de alguna forma los conflictos; en la práctica, el bloqueo parece inhibir la productividad más que otra cosa.

Hay una situación común donde el modelo bloquear-modificar-desbloquear resulta mejor, y es cuando tiene ficheros no-fusionables. Por ejemplo si el repositorio contiene algunas imágenes gráficas, y dos personas cambian la imagen a la vez, no hay forma de fusionar esos cambios.

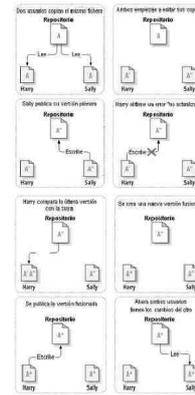


Ilustración 15: Solución Copiar-Modificar-Fusionar  
Fuente: tortoisesvn.net



### 1.2.4 Árbol, Rama y Etiqueta

#### 1.2.4.1 Árbol de Versiones

El árbol de versiones es la representación gráfica de las versiones de un archivo a través del tiempo.



Ilustración 16: Árbol de Revisiones

#### 1.2.4.2 Rama

Representa una bifurcación en un momento de tiempo de forma que, desde ese momento en adelante, dos copias de esos archivos puedan ser desarrolladas a diferentes velocidades o de diferentes formas, de modo independiente.

Una rama de una versión es útil para:

- Representar un camino independiente en el desarrollo.
- Representar diferentes variantes de un componente, las cuales pueden representar implementaciones en el tiempo/espacio, puertos de diferentes plataformas, etc.
- Representar desarrollo experimental.
- Permitir el desarrollo concurrente de varias personas las cuales unirán las fuentes al final, evitando las restricciones del desarrollo secuencial.

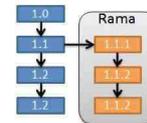


Ilustración 17: Rama

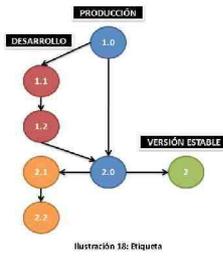
#### 1.2.4.3 Etiqueta

Consiste en un rotulo (nombre) de un fichero de un módulo en un momento en el tiempo.

Sirve para etiquetar los nodos del árbol de versiones.



Figura 14. Páginas 46-49 Libro “Análisis y Diseño de Sistemas 2”



**1.3 Nomenclatura de Versiones**

Los esquemas de versionamiento nos presentan algunas formas de versionar los productos de software para asegurara que cada versión tendrá un identificador único.

**1.3.1.1 Identificadores Basados en Secuencia**

En este esquema para versionar productos a cada release de software se le asocia un identificador único que consiste en una o más secuencias de números o letras.

Los cambios son clasificados por el nivel de significancia y la decisión de cual secuencia cambia entre releases es basada en la significancia de los cambios del release previo.

El número mayor se incrementa cuando existen saltos significantes en la funcionalidad, el número menor se incrementa cuando solamente algunas características menores han cambiado o cuando han sido agregados algunas reparaciones significativas. El número de revisión se incrementa cuando algunos errores menores son reparados.

El esquema de numeración es el siguiente:

- mayor.menor[.build[revisión]]
- O
- mayor.menor[mantenimiento.build]

**1.3.1.2 Fecha**

El proyecto Wine usa un modelo de versiones basado en fechas, el cual utiliza el año seguido por el mes, seguido por el día del release. Por ejemplo "Wine 20040505".

Cuando utilizamos fechas en versionamiento, es común utilizar el formato de fecha YYYY-MM-DD. Usualmente son omitidos los guiones.

**1.3.1.3 Año de Release**

Cuando una fecha es utilizada para denotar la versión, generalmente es para propósitos de mercadeo, y existe un número de versión interno. Por ejemplo Microsoft Windows 2000 Server es la versión NT 5.0 de Windows.

**1.3.1.4 Códigos Alfanuméricos**

Algunos productos utilizan códigos alfanuméricos para distinguir las versiones de sus releases, por ejemplo Macromedia Flash MX y Adobe Photoshop CS2.

**1.4 Administración del Cambio**

Entre los aspectos fundamentales de la administración de la configuración tenemos:

- Administración de las peticiones de cambio.
- Configuración de la contabilidad de estados.
- Administración de la configuración.

- Seguimiento del cambio.
- Selección de versión.
- Manufacturación del software.

**1.4.1.1 Administración de las Peticiones de Cambio**

Direciona la infraestructura organizacional requerida para evaluar los costos, cronograma, impacto de las peticiones de cambio para el producto existente.

**1.4.1.2 Configuración de la Contabilidad de Estados**

Es usado para describir el estado del producto basado en un tipo de defectos encontrados y corregidos durante el desarrollo del producto. Las métricas y la auditoría y los datos son importantes para determinar el estado completo del proyecto.

**1.4.1.3 Administración de la Configuración**

Describe la estructura del producto e identifica los elementos de su configuración, manteniendo las diferencias entre esas versiones.

Figura 15. Páginas 50-53 Libro “Análisis y Diseño de Sistemas 2”



Figura 16. Páginas 54-57 Libro “Análisis y Diseño de Sistemas 2”



**Puntos a recordar**

1. La administración de la configuración es la disciplina que se encarga de controlar la evolución de los sistemas de información a través de la construcción de diversas versiones.
2. El repositorio es una librería centralizada de archivos que provee el control de versiones en el repositorio.
3. El modelo checkout/checkin se enfoca en el soporte de versiones para archivos individuales. Los archivos son almacenados y versionados en el repositorio.
4. El Checkin ocurre cuando un cambio local es escrito en el repositorio.
5. El Checkout crea una copia local de una porción del repositorio.
6. La operación merge combina las modificaciones a un archivo que han ocurrido de forma independiente en dos ramas versionadas uniéndola en una nueva versión en una de las ramas.
7. El espacio de trabajo es una copia local del repositorio (o de una porción de él) que está sincronizada con el repositorio.
8. El árbol de versiones es la representación gráfica de las versiones de un archivo a través del tiempo.
9. Una rama representa una bifurcación en un momento de tiempo de forma que, desde ese momento en adelante, dos copias de esos archivos puedan ser desarrolladas.

**Artículos relacionados**

- **The Past, Present, and Future of Configuration Management**  
**Susan Dart**  
 Ref: [http://www.sei.cmu.edu/legacy/scm/abstracts/abscm\\_past\\_pres\\_future\\_TR08\\_92.html](http://www.sei.cmu.edu/legacy/scm/abstracts/abscm_past_pres_future_TR08_92.html)
- **Agile version control for multi-team development**  
**Henrik Kniberg**  
 Ref: <http://www.infoq.com/articles/agile-version-control>

## 8. DOCUMENTACIÓN DE APOYO DE “SOFTWARE AVANZADO”

A continuación se presenta el desarrollo de la documentación de apoyo del curso de *Software Avanzado*, para lo cual se realizó un libro que contiene las unidades más importantes impartidas a lo largo del curso.

### 8.1 Descripción general

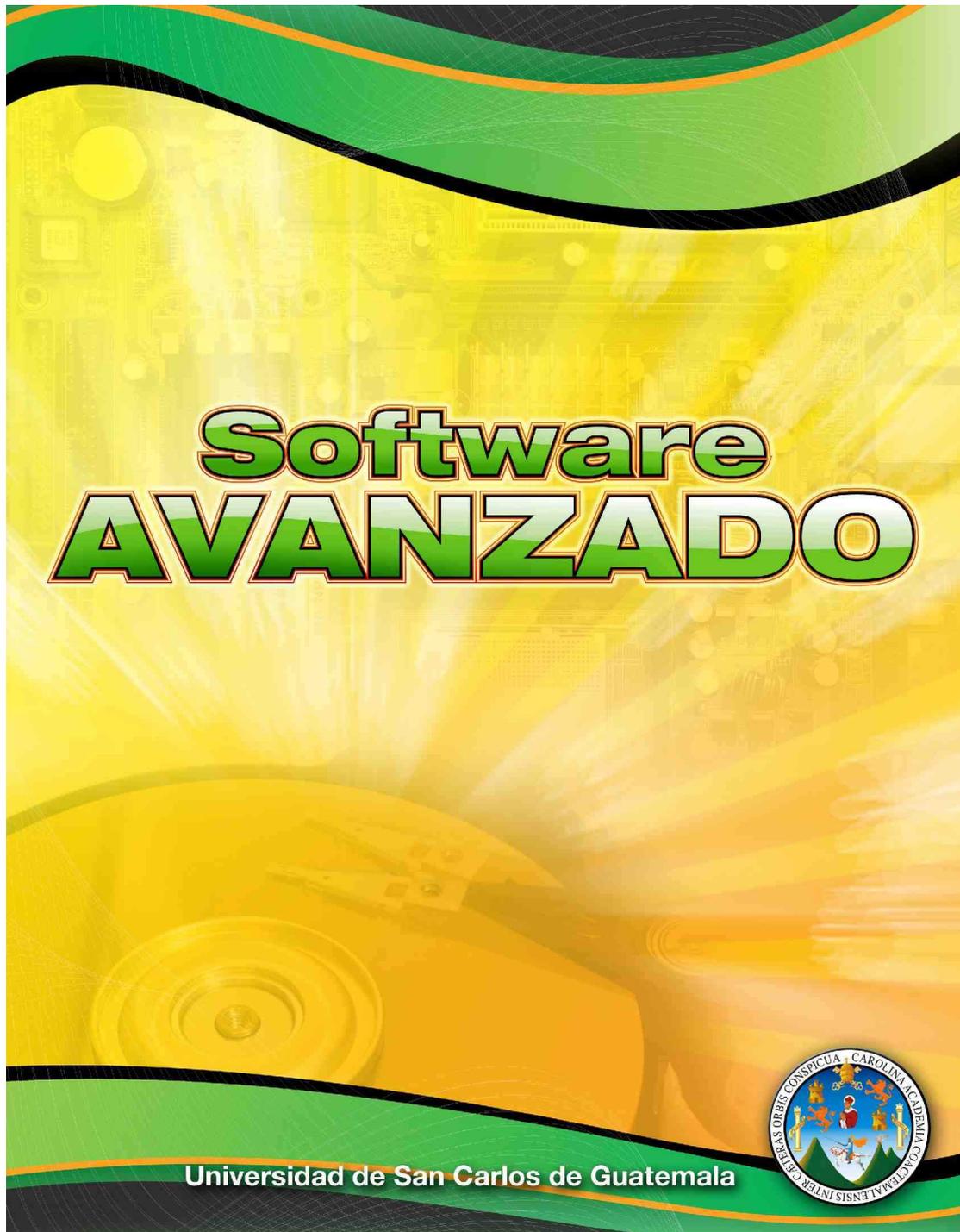
NOMBRE DE CURSO: *Software Avanzado* (780)

PRE- REQUISITOS: *Análisis y Diseño de Sistemas 2* (785)

POST – REQUISITOS: Ninguno

OBJETIVO: Este libro busca apoyar al estudiante en los conceptos relacionados con la administración de proyectos incluyendo calendarización, análisis de riesgos, cálculo del retorno de inversión a través de las metodologías propuestas por Microsoft *Solution Framework*.

Figura 17. Portada del Libro de *Software Avanzado*



## **8.2 *Microsoft Solutions Framework***

### **Descripción general**

Este capítulo presenta una breve introducción sobre el marco para gestión de proyectos *Microsoft Solutions Framework* y las siete disciplinas que dan soporte al proceso de gestión planteado por MSF.

### **Tabla de contenidos**

1. *Microsoft Solutions Framework*
  - 1.1. Principios de *Microsoft Solutions Framework*
2. Modelos de *Microsoft Solutions Framework*
  - 2.1. Modelo de equipo
    - 2.1.1. Roles en el modelo de equipo
    - 2.1.2. Beneficios del modelo de equipo
  - 2.2. Modelo de proceso
    - 2.2.1. Fases del modelo de proceso
    - 2.2.2. Beneficios del modelo de proceso
  - 2.3. Modelo de diseño de la solución
    - 2.3.1. Perspectivas del modelo
    - 2.3.2. Beneficios
  - 2.4. Modelo de aplicación
    - 2.4.1. Beneficios del modelo de aplicación
  - 2.5. Modelo de la arquitectura empresarial
    - 2.5.1. Perspectivas del modelo
    - 2.5.2. Beneficios del modelo de arquitectura empresarial
  - 2.6. Modelo de infraestructura
    - 2.6.1. Beneficios del modelo de infraestructura

## 2.7. Modelo de costo total de propiedad

### 2.7.1. Beneficios del modelo de costo total de propiedad

## 8.3 Modelo de equipo

### Descripción general

Este capítulo describe los aspectos fundamentales del modelo de equipo de Microsoft *Solutions Framework* para la construcción de equipos de trabajo fuertes y eficientes. Asimismo se describe la función de cada uno de los roles definidos en el modelo y los principios que lo fundamentan.

### Tabla de contenidos

1. Modelo de equipo
2. Fundamentos
3. Principios
4. Roles
  - 4.1. Administrador del producto
    - 4.1.1. Áreas funcionales del administrador del producto
  - 4.2. Administrador del programa
    - 4.2.1. Áreas funcionales del administrador del programa
  - 4.3. Desarrollador
    - 4.3.1. Áreas funcionales del desarrollador
  - 4.4. *Tester*
    - 4.4.1. Áreas funcionales del *tester*
  - 4.5. Capacitador de usuarios
    - 4.5.1. Áreas funcionales del capacitador de usuarios

#### 4.6. Logística

##### 4.6.1. Áreas funcionales de logística

## 8.4 Modelo de proceso

### Descripción general

Este capítulo presenta una panorámica general del modelo de proceso de Microsoft *Solutions Framework* e incluye una breve descripción sobre cada una de las fases e hitos del modelo.

### Tabla de contenidos

1. Modelo de proceso
2. Fases del modelo de proceso
  - 2.1. Fase de visionado
    - 2.1.1. Declaración del problema
    - 2.1.2. Declaración de la visión
    - 2.1.3. El alcance
    - 2.1.4. Perfiles de usuario
    - 2.1.5. Tareas asignadas a los distintos roles durante la fase de visionado
  - 2.2. Fase de planificación
    - 2.2.1. Análisis de los requerimientos del negocio
    - 2.2.2. Retorno de inversión
    - 2.2.3. Modelo de costo total de propiedad
    - 2.2.4. Gestión del riesgo
    - 2.2.5. Análisis funcional
    - 2.2.6. Cronograma del proyecto

- 2.2.7. Tareas asignadas a los distintos roles durante la fase de planificación
- 2.3. Fase de desarrollo
  - 2.3.1. Tareas asignadas a los distintos roles durante la fase de desarrollo
- 2.4. Fase de estabilización
  - 2.4.1. Tareas asignadas a los distintos roles durante la fase de estabilización
- 2.5. Fase de despliegue

## **8.5 Disciplinas de gestión de proyectos y gestión de riesgos**

### **Descripción general**

Este capítulo presenta una breve introducción a las disciplinas de gestión de riesgo y gestión de proyectos de Microsoft *Solutions Framework*, las cuales constituyen unas de las principales herramientas para la gestión de proyectos existentes en el mercado.

### **Tabla de contenidos**

- 1. Disciplina de gestión de proyectos
  - 1.1. El espectro de la gestión de proyectos
    - 1.1.1. Las personas
    - 1.1.2. El producto
    - 1.1.3. El proceso
    - 1.1.4. El proyecto
  - 1.2. Las señales de peligro de un proyecto de *software*

- 1.3. El principio W5HH de Barry Bohem
- 1.4. Áreas de la gestión de proyectos
- 1.5. Características de la gestión de proyectos de MSF
  - 1.5.1. El Rol de administrador de proyectos está inmerso en la gestión de proyectos
  - 1.5.2. El gestor de proyectos y los líderes de equipos
  - 1.5.3. Responsabilidades de la gestión de proyectos
- 1.6. Recomendaciones para los equipos
  - 1.6.1. Administración del alcance
  - 1.6.2. Preparar planes
  - 1.6.3. Reutilización de documentos
  - 1.6.4. Planes de proyecto
  - 1.6.5. *Work breakdown structure*
  - 1.6.6. Estimación en proyectos de *software*
- 2. Disciplina de gestión del riesgo
  - 2.1. Mitigación del riesgo
  - 2.2. Niveles de la gestión de riesgo
  - 2.3. Integrar el análisis de riesgo al ciclo de vida de un proyecto
  - 2.4. Proceso de la gestión de riesgo
    - 2.4.1. Identificar el riesgo
    - 2.4.2. Analizar y priorizar riesgos
    - 2.4.3. Planificar y calendarizar el riesgo
    - 2.4.4. Seguimiento y reporte del estado del riesgo
    - 2.4.5. Control de riesgo
    - 2.4.6. Aprender del riesgo
  - 2.5. Análisis costo beneficio de la gestión de riesgo

## 8.6 Métricas de proyectos

### Descripción general

En este capítulo encontrará una breve descripción sobre el proceso de toma de métricas y estimación de tiempos de proyectos, basados en los métodos de líneas de código y puntos de función.

### Tabla de contenidos

1. Estimación
2. Métricas
3. Métricas en el proceso
4. Métricas de proyecto
5. Mediciones de *software*
  - 5.1. Líneas de código
  - 5.2. Puntos de función

## **8.7 Costo total de propiedad y retorno de inversión**

### **Descripción general**

En este capítulo encontrará una breve introducción al modelo de costo total de propiedad de MSF y los métodos para cálculo del retorno de inversión para un proyecto de tecnología.

### **Tabla de contenidos**

1. Modelo de costo total de propiedad
  - 1.1. Beneficios
  - 1.2. Costos
  - 1.3. Consideraciones a tomar en cuenta
  - 1.4. Ciclo de vida
    - 1.4.1. Fase de análisis
    - 1.4.2. Fase de mejora del TCO
    - 1.4.3. Fase de administración del costo total de propiedad
2. Retorno de inversión
  - 2.1. Métodos para el análisis de retorno de inversión
    - 2.1.1. Beneficios
    - 2.1.2. Costos
    - 2.1.3. Razón costo beneficio
    - 2.1.4. Retorno de inversión
    - 2.1.5. Valor actual neto
    - 2.1.6. Punto de equilibrio

## 8.8 Demo: capítulo 1 “Microsoft Solutions Framework”

Figura 18. Páginas 10-13 Libro “Software Avanzado”

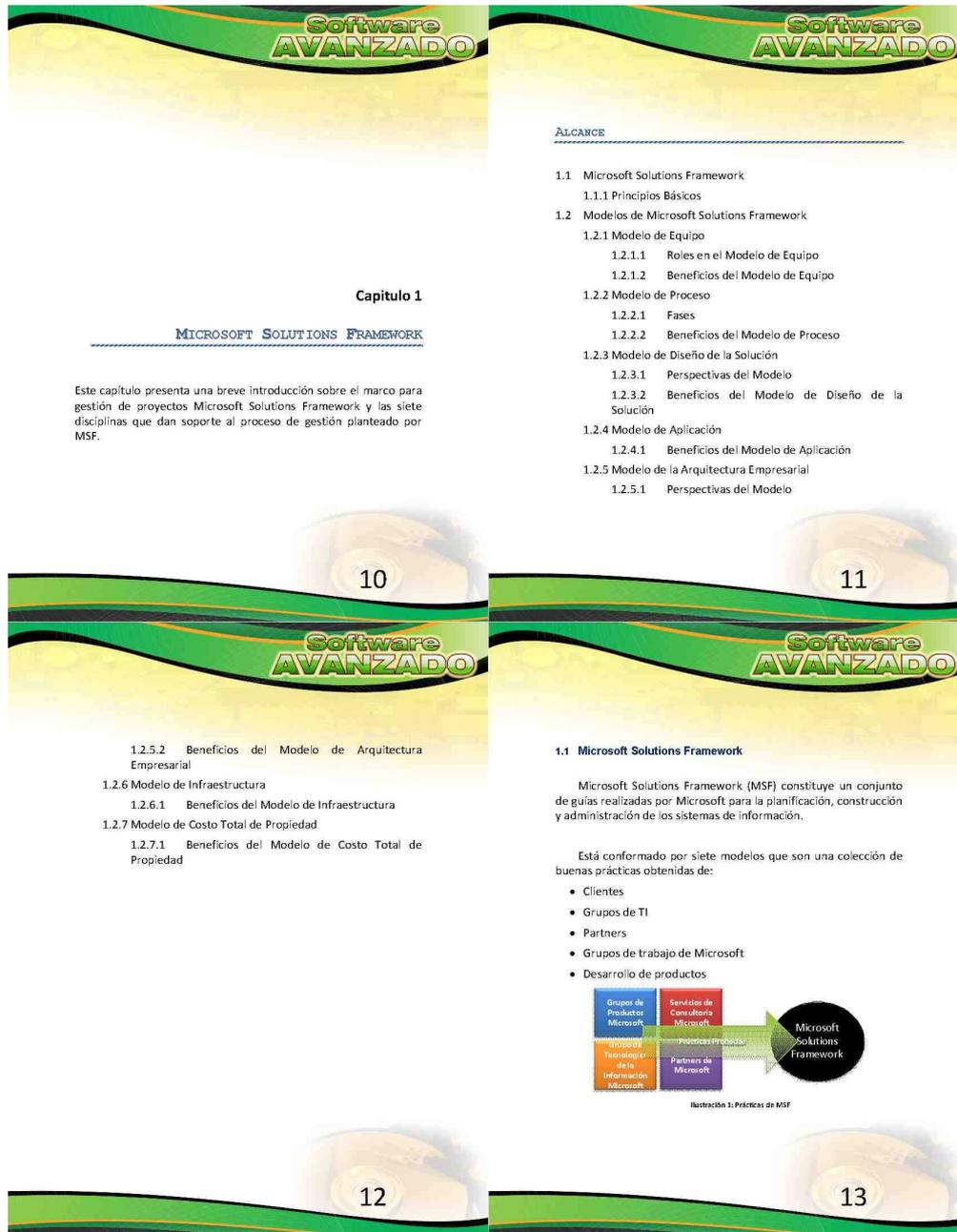


Figura 19. Páginas 14-17 Libro “Software Avanzado”

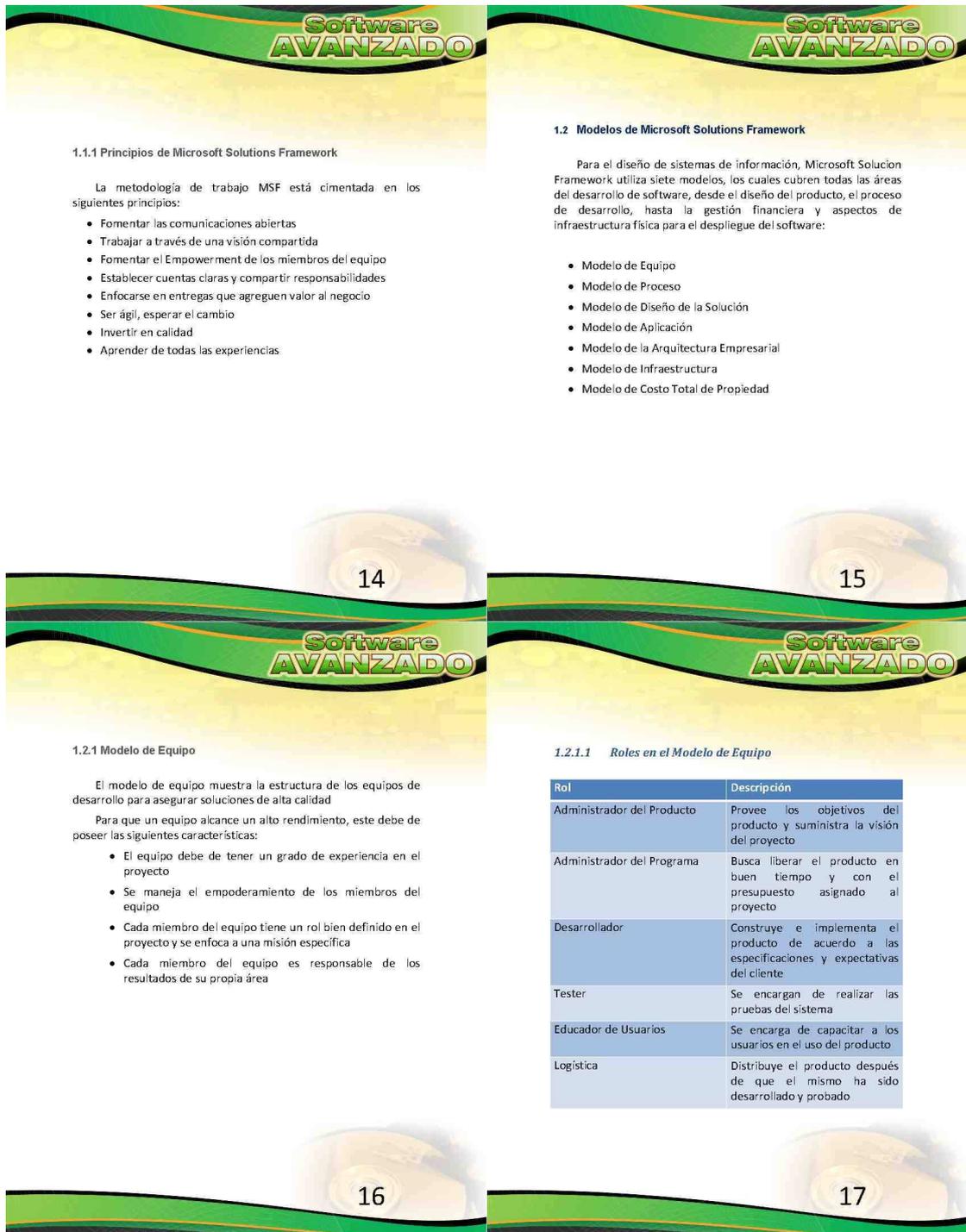


Figura 20. Páginas 18-21 Libro “Software Avanzado”

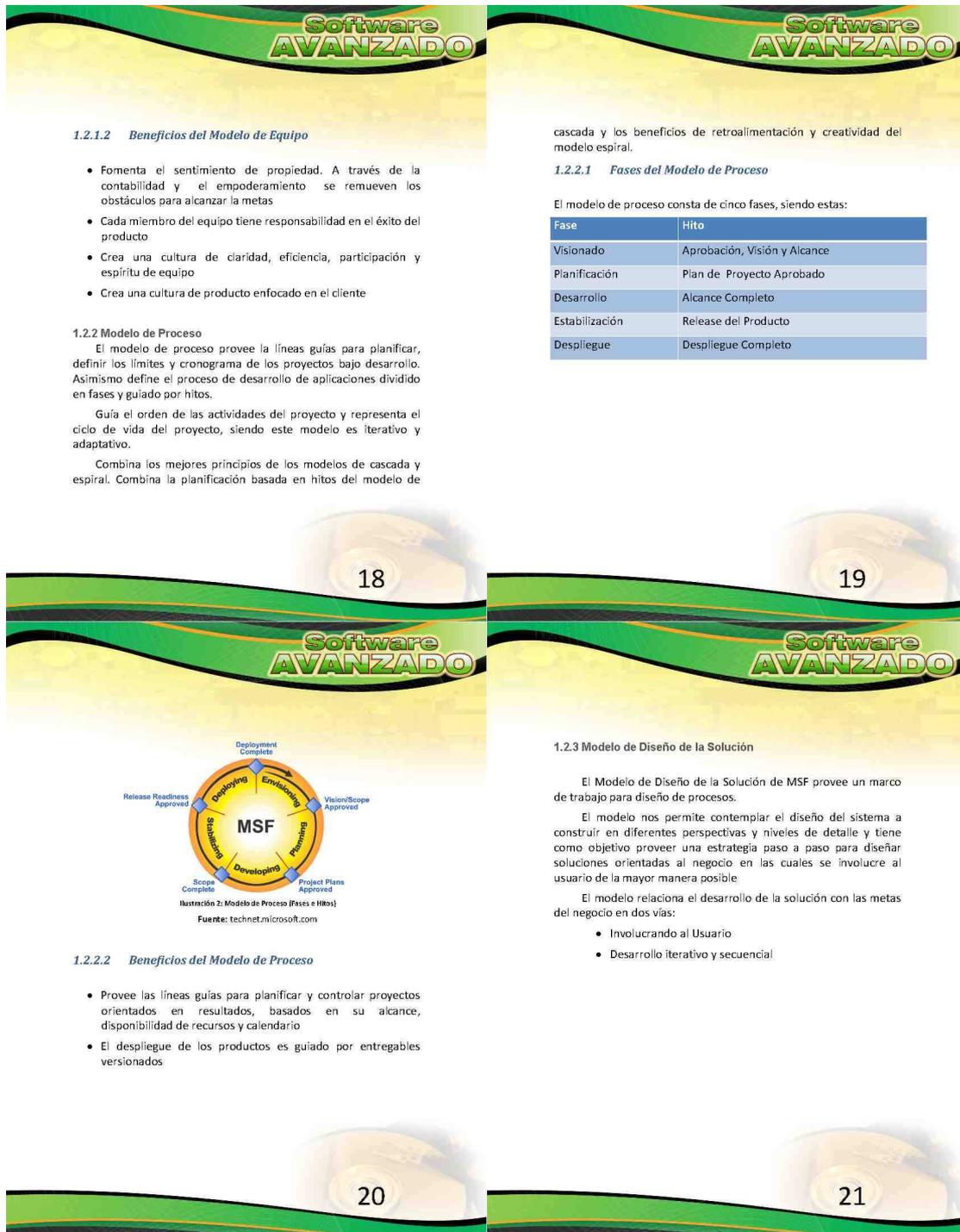
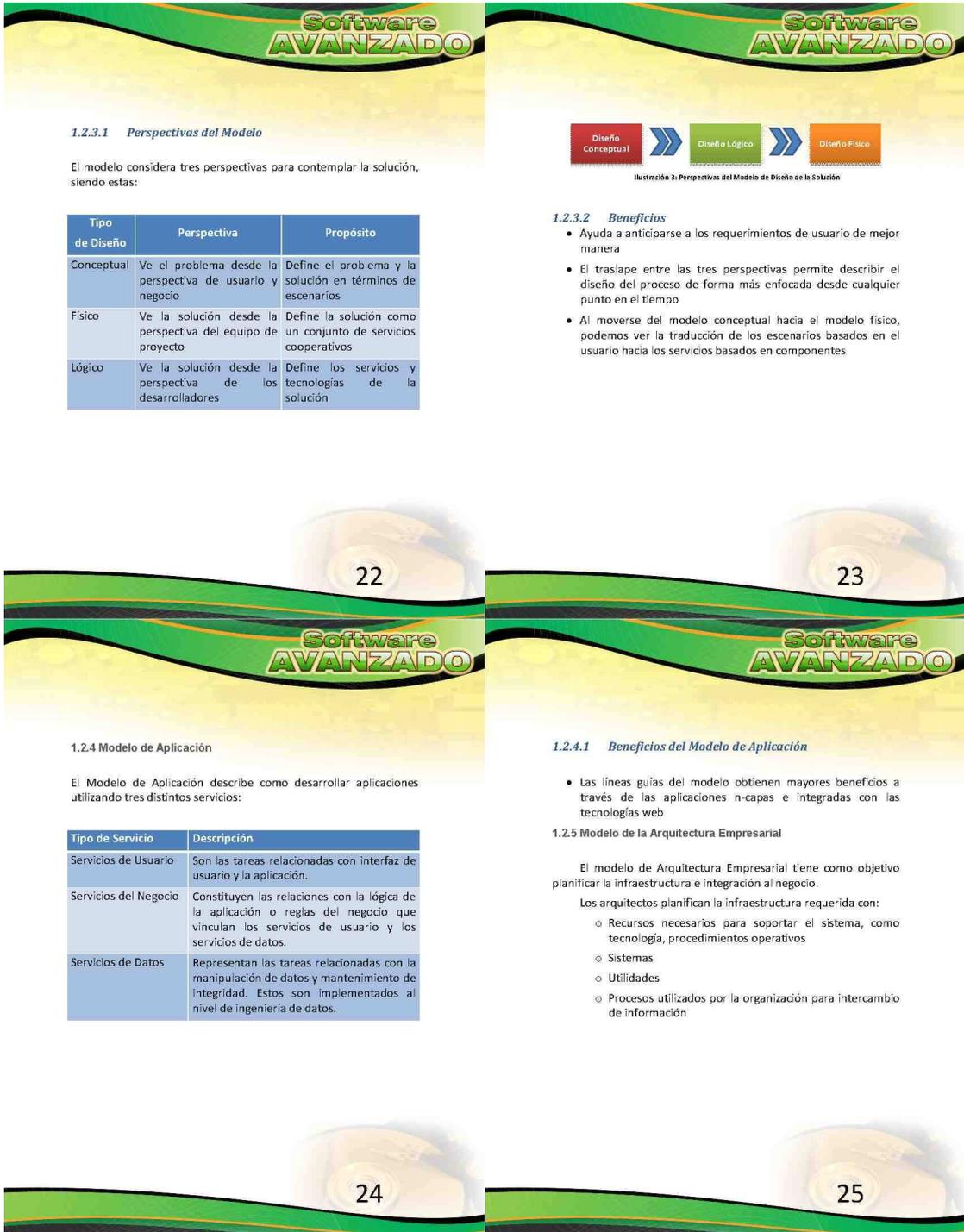


Figura 21. Páginas 22-25 Libro “Software Avanzado”



1.2.3.1 Perspectivas del Modelo

El modelo considera tres perspectivas para contemplar la solución, siendo estas:

Tipo de Diseño	Perspectiva	Propósito
Conceptual	Ve el problema desde la perspectiva de usuario y negocio	Define el problema y la solución en términos de escenarios
Físico	Ve la solución desde la perspectiva del equipo de proyecto	Define la solución como un conjunto de servicios cooperativos
Lógico	Ve la solución desde la perspectiva de los desarrolladores	Define los servicios y tecnologías de la solución



Ilustración 3: Perspectivas del Modelo de Diseño de la Solución

1.2.3.2 Beneficios

- Ayuda a anticiparse a los requerimientos de usuario de mejor manera
- El traslape entre las tres perspectivas permite describir el diseño del proceso de forma más enfocada desde cualquier punto en el tiempo
- Al moverse del modelo conceptual hacia el modelo físico, podemos ver la traducción de los escenarios basados en el usuario hacia los servicios basados en componentes

1.2.4 Modelo de Aplicación

El Modelo de Aplicación describe como desarrollar aplicaciones utilizando tres distintos servicios:

Tipo de Servicio	Descripción
Servicios de Usuario	Son las tareas relacionadas con interfaz de usuario y la aplicación.
Servicios del Negocio	Constituyen las relaciones con la lógica de la aplicación o reglas del negocio que vinculan los servicios de usuario y los servicios de datos.
Servicios de Datos	Representan las tareas relacionadas con la manipulación de datos y mantenimiento de integridad. Estos son implementados al nivel de ingeniería de datos.

1.2.4.1 Beneficios del Modelo de Aplicación

- Las líneas guías del modelo obtienen mayores beneficios a través de las aplicaciones n-capas e integradas con las tecnologías web

1.2.5 Modelo de la Arquitectura Empresarial

El modelo de Arquitectura Empresarial tiene como objetivo planificar la infraestructura e integración al negocio.

- Los arquitectos planifican la infraestructura requerida con:
- Recursos necesarios para soportar el sistema, como tecnología, procedimientos operativos
  - Sistemas
  - Utilidades
  - Procesos utilizados por la organización para intercambio de información

Figura 22. Páginas 26-29 Libro “Software Avanzado”

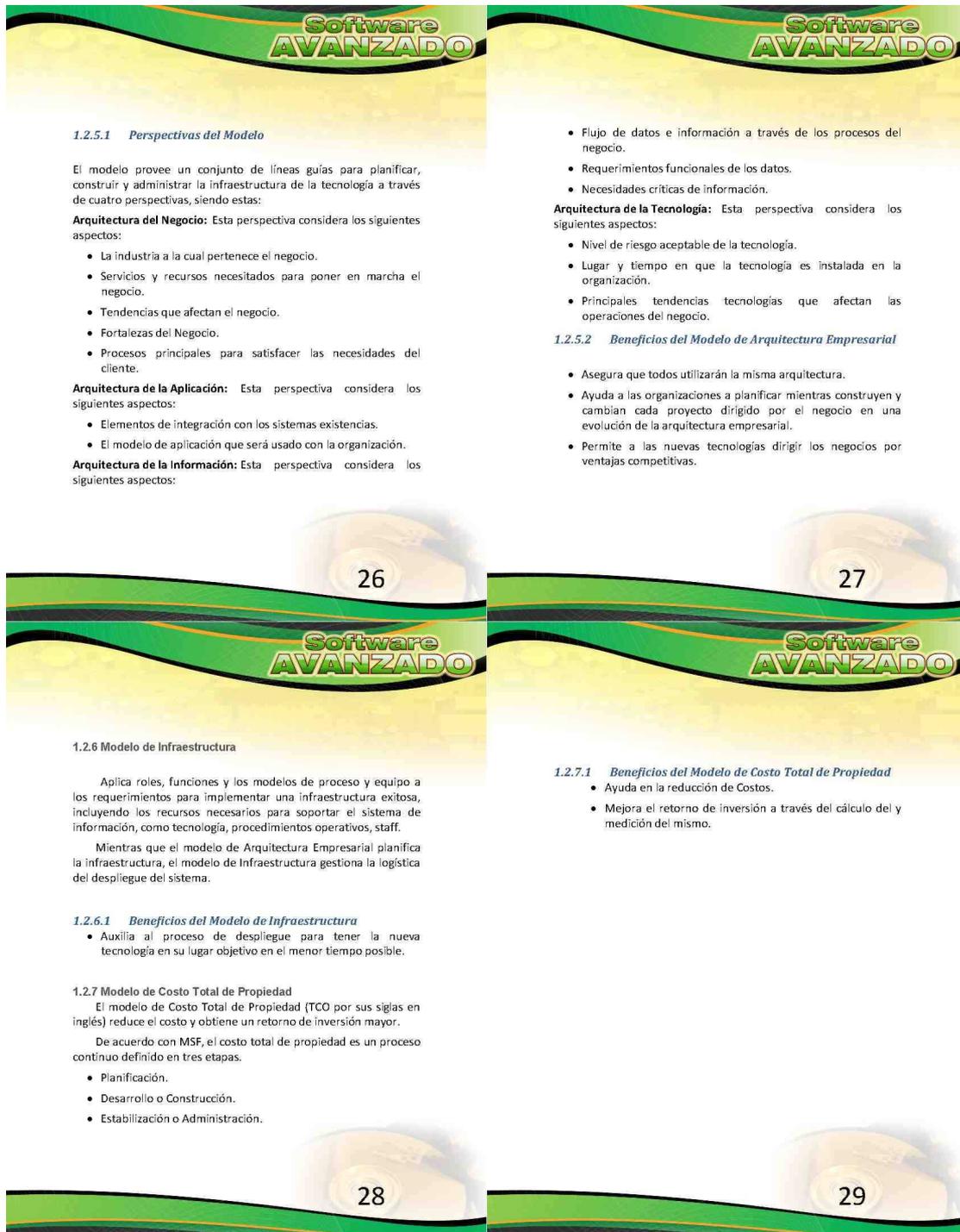


Figura 23. Páginas 30-31 Libro “Software Avanzado”





## CONCLUSIONES

1. Se creó una herramienta para definición, actualización, evaluación y promoción de laboratorios de cursos denominada “Guía del Instructor”.
2. Se estructuraron las guías de instructor de los laboratorios de los cursos Análisis y Diseño de Sistemas 1, Análisis y Diseño de Sistemas 2 y *Software Avanzado*. El contenido que se incluyó en cada laboratorio fue sugerido y coordinado por los catedráticos titulares de los cursos en cuestión.
3. Para cada guía de instructor se desarrollaron los siguientes elementos: Información general del curso, distribución de horas y actividades, forma de evaluación, detalle de sesiones, distribución de sesiones, detalle de tareas asignadas, detalle de tutoriales, detalle de exámenes, ejemplos de exámenes, bibliografía y anexos.
4. Se desarrolló el contenido de los laboratorios en presentaciones, donde cada presentación constituye una sesión teórica del laboratorio.

5. Para las prácticas, exámenes y tutoriales de laboratorios, se definieron los objetivos que se buscan de cada actividad y se realizó un ejemplo de cada uno para hacer más sencilla su comprensión por parte del implementador de la guía de laboratorio.
6. Se realizó un libro para los cursos Sistemas de Análisis y Diseño de Sistemas 1, Análisis y Diseño de Sistemas 2 y Software Avanzado. El contenido fue estructurado con base a los programas de curso y actualizado en un esfuerzo en conjunto con los catedráticos titulares de los cursos mencionados anteriormente.
7. Al final de cada unidad de las guías de curso se incluyen artículos de interés como apoyo al contenido base.
8. Mediante los entregables del proyecto de ESTRUCTURACIÓN DE LOS LABORATORIOS Y DOCUMENTACIÓN DE APOYO DE LOS CURSOS: ANÁLISIS Y DISEÑO DE SISTEMAS 1, ANÁLISIS Y DISEÑO DE SISTEMAS 2, SOFTWARE AVANZADO APLICANDO LA METODOLOGÍA DEL IT CENTER INDIA-GUATEMALA, DE LA CARRERA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA se conformó una guía detallada para impartir los laboratorios y clases de los cursos anteriormente mencionados de la carrera de Ingeniería en Ciencias y Sistemas. Se provee la plataforma para estandarizar los contenidos y para facilitar la actualización de los mismos.

## RECOMENDACIONES

1. Las guías de instructor deben de ser actualizadas de forma constante en periodos no mayores a un año. La estructura de la guía debe de mantenerse, modificando únicamente el contenido de cada guía.
2. Debe de desarrollarse una planificación semestral por parte de los auxiliares de curso para ajustar las sesiones teóricas y prácticas de la guía del instructor a la duración del semestre y definir de forma previa la fecha de realización de actividades y sesiones teóricas.
3. Se recomienda almacenar en un repositorio de documentos las diversas versiones de las guías de instructor, utilizadas por la Escuela de Ingeniería en Ciencias y Sistemas, para formar una base de datos de conocimiento y llevar el seguimiento del avance y evolución de los cursos de la carrera.



## BIBLIOGRAFÍA

1. Abran, Alain y Moore, James. **Guide to the software engineering body of knowledge**. IEEE Computer Society, 2004. 200 pp.
2. Berczuk, Stephen P. y Appleton, Brad. **Software Configuration Management Patterns: Effective Teamwork, Practical Integration**. Estados Unidos de Norte América: Editorial Addison-Wesley, 2003. 218 pp.
3. Chang, James F. **Business process management systems: strategy and implementation**. Estados Unidos de Norteamérica: Editorial Auerbach Publications, 2006. 286 pp.
4. Cohn, Mike. **User Stories Applied: For Agile Software Development**. Estados Unidos de Norteamérica: Editorial Addison Wesley, 2004. 304 pp.
5. Collins-Sussman, Ben y otros. **Version Control with Subversion For Subversion 1.2**, 2006.
6. Crawford, William y Kaplan, Jonathan. **J2EE Design Patterns**. Estados Unidos de Norte América: Editorial O'Reilly, 2003.
7. Dart, Susan. **Concepts in Configuration Management Systems**. Artículo. Estados Unidos de Norteamérica, universidad Carneige Mellon, Software Engineering Institute.

8. Erl, Thomas. **Service-Oriented Architecture: Concepts, Technology, and Design.** Prentice Hall Professional Technical Reference, 2005. 792 pp.
9. Freeman, Eric y Freeman, Elisa. **Head First, Design Patterns.** Estados Unidos de Norte América: Editorial O'Reilly, 2004. 638 pp.
10. F. Rico, David. **ROI of Software Process Improvement: Metrics for Project Managers and Software Engineers.** Editorial J. Ross Publishing, 2004. 218 pp.
11. Gamma, Erich, y otros. **Patrones de Diseño, Elementos de software orientado a objetos reutilizable.** Estados Unidos de Norteamérica: Addison Wesley, 2003.
12. Gorton, Ian. **Essential Software Architecture.** Estados Unidos de Norte América: Editorial Springer, 2006. 283 pp.
13. Havey, Mike. **Essential Business Process Modeling.** Editorial O'Reilly, 2005. 350 pp.
14. Hohmann, Luke. **Beyond Software Architecture: Creating and Sustaining Winning Solutions.** Estados Unidos de Norte América: Editorial Pearson Education, 2003.
15. Johnson, Rod y otros. **Spring java/j2ee Application Framework 2.0.**
16. Kroll, Per y Kruchten, Philippe. **The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP.** Estados

- Unidos de Norteamérica: Editorial Addison Wesley, 2003. 464 pp.
17. Leavitt, Michael y Shneiderman, Ben. **Research-Based Web Design & Usability Guideline**. Gobierno de los Estados Unidos de Norteamérica, 2001.
  18. McLaughlin, Brett y otros. **Head First, Object-Oriented Analysis & Design**. Estados Unidos de Norte América: Editorial O'Reilly, 2006. 600 pp.
  19. Microsoft Corporation. **Analyzing Requirements and Defining Microsoft .Net Solution Architectures**. Estados Unidos de Norteamérica: Editorial Microsoft Press, 2003.
  20. Mrutunjaya Panda, **Curso de Planificación Educativa a través de Instructor Guidelines**. India-Guatemala IT Education Centre of Excellence. Guatemala. Enero a julio de 2008.
  21. Paul M. Duvall. **Continuous Integration: Improving Software Quality and Reducing Risk**. Estados Unidos de Norteamérica: Editorial Addison-Wesley, 2007. 283 pp.
  22. Presman, Roger. **Software Engineering, a Practitioner's Approach**. 5ª ed. Editorial McGraw Hill, 2001. 860 pp.
  23. V. Turner, Michael S. **Microsoft Solutions Framework Essentials: Building Successful Technology Solutions**. Editorial Microsoft Press, 2006. 336 pp.

24. Weske, Mathias. **Business Process Management, Concepts, Languages, Architectures.** Editorial Springer, 2007. 368 pp.
25. Wigodsky, Andrew S. **RAPID Value Management for the Business Cost of Ownership: Readiness, Architecture, Process, Integration, Deployment.** Estados Unidos de Norteamérica: Editorial Digital Press, 2004, 297 pp.

## ANEXOS

Para la estructuración de los laboratorios y el contenido de la documentación de apoyo se contó con el soporte y revisión de los catedrático responsables de cada uno de los cursos en la Escuela de Ciencias y Sistemas.

Por lo cual, en el anexo se adjuntan las cartas autorizando el contenido de los laboratorios por parte del catedrático titular de cada curso:

- Ing. Ricardo Morales Prado – Análisis y Diseño de Sistemas 1.
- Ing. Ricardo Morales Prado – Análisis y Diseño de Sistemas 2.
- Ing. Julio Contreras – Software Avanzado.

Para la documentación de apoyo de los cursos el catedrático titular de cada curso sugirió el contenido de cada documento que se desarrolló. En este proceso participaron los siguientes catedráticos:

- Ing. Ricardo Morales Prado – Análisis y Diseño de Sistemas 1.
- Ing. Ricardo Morales Prado – Análisis y Diseño de Sistemas 2.
- Ing. Luis Espino – Software Avanzado.

Donde se revisó todos los documentos necesarios y a criterio de cada catedrático estos cumplen con el objetivo y contenido planteado para el laboratorio y documentación de cada curso.





Guatemala 07 de octubre de 2008

Ingeniero  
Jorge Armin Mazariegos  
Asesor de EPS  
Facultad de Ingeniería  
USAC

Por este medio me dirijo a usted para informarle que he revisado el **Guideline de "Análisis y Diseño de Sistemas I"**, del Proyecto "Estructuración de los laboratorios y documentación de apoyo de los cursos Análisis y Diseño de Sistemas I, Análisis y Diseño de Sistemas II y Software Avanzado", y a mi parecer cumple con los objetivos de guía de laboratorio.

Agradeciendo su atención a la presente,

Atentamente,

  
\_\_\_\_\_  
Ing. Ricardo Morales Prado  
Catedrático de la Escuela de Ciencias y Sistemas  
Facultad de Ingeniería, USAC

  
V.B.





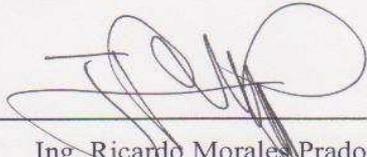
Guatemala 07 de octubre de 2008

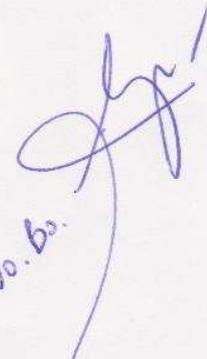
Ingeniero  
Jorge Armin Mazariegos  
Asesor de EPS  
Facultad de Ingeniería  
USAC

Por este medio me dirijo a usted para informarle que he revisado el **Guideline de "Análisis y Diseño de Sistemas 2"**, del Proyecto "Estructuración de los laboratorios y documentación de apoyo de los cursos Análisis y Diseño de Sistemas I, Análisis y Diseño de Sistemas II y Software Avanzado", y a mi parecer cumple con los objetivos de guía de laboratorio.

Agradeciendo su atención a la presente,

Atentamente,

  
\_\_\_\_\_  
Ing. Ricardo Morales Prado  
Catedrático de la Escuela de Ciencias y Sistemas  
Facultad de Ingeniería, USAC

  
Vo. bo.





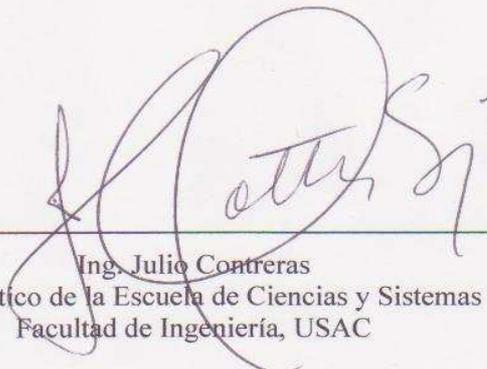
Guatemala 9 de septiembre de 2008

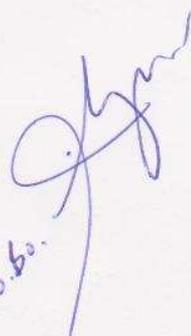
Ingeniero  
Jorge Armin Mazariegos  
Asesor de EPS  
Facultad de Ingeniería  
USAC

Por este medio me dirijo a usted para informarle que he revisado el **Guideline de "Software Avanzado"**, del Proyecto "Estructuración de los laboratorios y documentación de apoyo de los cursos Análisis y Diseño de Sistemas I, Análisis y Diseño de Sistemas II y Software Avanzado", y a mi parecer cumple con los objetivos de guía de laboratorio.

Agradeciendo su atención a la presente,

Atentamente,

  
\_\_\_\_\_  
Ing. Julio Contreras  
Catedrático de la Escuela de Ciencias y Sistemas  
Facultad de Ingeniería, USAC

  
V.O. B.O.





Guatemala 27 de mayo de 2009

Ingeniero  
Jorge Armin Mazariegos  
Asesor de EPS  
Facultad de Ingeniería  
USAC

Por este medio me dirijo a usted para informarle que he revisado el **“Material de Apoyo del curso *Análisis y Diseño de Sistemas 1*”**, del Proyecto “Estructuración de los laboratorios y documentación de apoyo de los cursos Análisis y Diseño de Sistemas I, Análisis y Diseño de Sistemas II y Software Avanzado”, y a mi parecer cumple con los objetivos del curso.

Agradeciendo su atención a la presente,

Atentamente,

---

Ing. Ricardo Morales Prado  
Catedrático de la Escuela de Ciencias y Sistemas  
Facultad de Ingeniería, USAC





Guatemala, 5 de Septiembre de 2009

Ingeniero  
Jorge Armin Mazariegos  
Asesor de EPS  
Facultad de Ingeniería  
USAC

Por este medio me dirijo a usted para informarle que he revisado el **“Material de Apoyo del curso “Análisis y Diseño de Sistemas 2”**, del Proyecto “Estructuración de los laboratorios y documentación de apoyo de los cursos Análisis y Diseño de Sistemas I, Análisis y Diseño de Sistemas II y Software Avanzado”, y a mi parecer cumple con los objetivos del curso.

Agradeciendo su atención a la presente,

Atentamente,

---

Ing. Ricardo Morales Prado  
Catedrático de la Escuela de Ciencias y Sistemas  
Facultad de Ingeniería, USAC





Guatemala 21 de enero de 2009

Ingeniero  
Jorge Armin Mazariegos  
Asesor de EPS  
Facultad de Ingeniería  
USAC

Por este medio me dirijo a usted para informarle que he revisado el **Libro "Material de Apoyo Software Avanzado"**, del Proyecto "Estructuración de los laboratorios y documentación de apoyo de los cursos Análisis y Diseño de Sistemas I, Análisis y Diseño de Sistemas II y Software Avanzado", y a mi parecer cumple con los objetivos de guía de laboratorio.

Agradeciendo su atención a la presente,

Atentamente,

---

Ing. Luis Espino  
Catedrático de la Escuela de Ciencias y Sistemas  
Facultad de Ingeniería, USAC