

INGENIERIA EN TECNOLOGIAS DE LA INFORMACION

HOJA DE ASIGNATURA CON DESGLOSE DE UNIDADES TEMÁTICAS

1. Nombre de la asignatura	Optativa II: Modelos de Proceso para el Desarrollo de Software.
2. Competencias	<ul style="list-style-type: none">Dirigir proyectos de tecnologías de información (T.I.) para contribuir a la productividad y logro de los objetivos estratégicos de las organizaciones utilizando las metodologías apropiadas.
3. Cuatrimestre	Tercero
4. Horas Prácticas	55
5. Horas Teóricas	35
6. Horas Totales	90
7. Horas Totales por Semana Cuatrimestre	6
8. Objetivo de la Asignatura	El alumno realizará el modelado de un desarrollo de software con base en las especificaciones de requerimientos, diseñará y aplicará pruebas para evaluar el producto.

Unidades Temáticas	Horas		
	Prácticas	Teóricas	Totales
I. Introducción a la ingeniería de software.	5	5	10
II. Diagramas UML.	20	5	25
III. Ingeniería de requerimientos.	10	5	15
IV. Modelos de proceso.	15	10	25
V. Pruebas y aseguramiento de la calidad.	10	5	15
Totales	55	35	90

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

UNIDADES TEMÁTICAS

1. Unidad Temática	I. Introducción a la Ingeniería de software
2. Horas Prácticas	5
3. Horas Teóricas	5
4. Horas Totales	10
5. Objetivo	El alumno conocerá los principios y metodologías para el desarrollo y mantenimiento de sistemas de información.

Temas	Saber	Saber hacer	Ser
1. Definición de ingeniería de software y su importancia.	<ul style="list-style-type: none"> Definir el concepto de ingeniería de software. 	<ul style="list-style-type: none"> Comprender claramente el concepto de Ingeniería de software y su importancia. 	<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Coherente. Discreto. Proactivo. Asertivo. Hábil para comunicarse efectivamente. Liderazgo. Hábil para trabajo en equipo.
2. Ciclo de Vida de un Sistema Software.	<ul style="list-style-type: none"> Definir el concepto Ciclo de Vida del Software. 	<ul style="list-style-type: none"> Enumerar los elementos del ciclo de vida de software Elegir el ciclo de vida adecuado de acuerdo a las características de los proyectos. 	<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Coherente. Discreto. Proactivo. Asertivo. Hábil para comunicarse efectivamente. Liderazgo. Hábil para trabajo en equipo.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

Proceso de evaluación		
Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<ul style="list-style-type: none"> • El alumno elaborará un análisis de los diferentes ciclos de vida software, resaltando sus principales características, beneficios y desventajas. 	<ol style="list-style-type: none"> 1. Identificar el concepto de Ingeniería de Software. 2. Identificar los elementos para diseñar, construir y entregar software. 3. Identificar los diferentes ciclos de vida software. 	<ul style="list-style-type: none"> • Ejercicios prácticos. • Lista de cotejo. • Relación y definición de conceptos. • Evaluación de casos.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

Proceso enseñanza aprendizaje	
Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none"> • Proyectos de Investigación. • Solución de problemas. • Estudios de Casos. • Aprendizaje basado en Proyectos. • instrucción directa. 	<ul style="list-style-type: none"> • Cañón. • Equipo de Cómputo. • Pintarrón. • Plumones.

Espacio Formativo		
Aula	Laboratorio / Taller	Empresa
X		

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

UNIDADES TEMÁTICAS

1. Unidad Temática	II. Diagramas UML
2. Horas Prácticas	20
3. Horas Teóricas	5
4. Horas Totales	25
5. Objetivo	El alumno utilizará el lenguaje UML para representar la estructura y el funcionamiento de un sistema.

Temas	Saber	Saber hacer	Ser
1.Introducción a UML.	<ul style="list-style-type: none"> Definir UML Identificar su uso. 		<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Asertivo. Hábil para trabajo en equipo.
2.Representación de la estructura del sistema por medio de UML.	<ul style="list-style-type: none"> Identificar la notación UML para representar los aspectos estructurales o estáticos de un sistema. 	<ul style="list-style-type: none"> Elaborar los principales diagramas UML para representar la estructura sistema. 	<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Asertivo. Hábil para trabajo en equipo.
3.Representación del comportamiento del sistema por medio de UML.	<ul style="list-style-type: none"> Identificar la notación UML para representar el comportamiento de un sistema. 	<ul style="list-style-type: none"> Elaborar los diagramas UML para representar el comportamiento de un sistema. 	<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Asertivo. Hábil para trabajo en equipo.
4.Representación de la interacción del sistema por medio de UML.	<ul style="list-style-type: none"> Identificar la notación UML para representar la interacción en un sistema. 	<ul style="list-style-type: none"> Elaborar los principales diagramas UML para representar la interacción en un sistema. 	<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Asertivo. Hábil para trabajo en equipo.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

Proceso de evaluación		
Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<ul style="list-style-type: none"> • El alumno presentará un documento de análisis y diseño del sistema a través de diagramas en UML. 	<ol style="list-style-type: none"> 1. Identificar el concepto de UML. 2. Identificar los principales diagramas de UML para representar la estructura de un sistema. 3. Identificar los principales diagramas UML para presentar el comportamiento de un sistema. 4. Identificar los principales diagramas UML para representar la interacción de un sistema. 5. Representar un sistema utilizando notación UML. 	<ul style="list-style-type: none"> • Ejecución de Tareas. • Listas de verificación. • Desarrollo de Productos.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

Proceso enseñanza aprendizaje	
Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none"> • Ejercicios prácticos. • Estudios de Casos. • Aprendizaje basado en Proyectos. • Instrucción directa. 	<ul style="list-style-type: none"> • Cañón. • Equipo de Cómputo. <li style="background-color: yellow;">• Software de Modelado UML. • Pintarrón. • Plumones.

Espacio Formativo		
Aula	Laboratorio / Taller	Empresa
X	X	

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

UNIDADES TEMÁTICAS

1. Unidad Temática	III. Ingeniería de requerimientos
2. Horas Prácticas	10
3. Horas Teóricas	5
4. Horas Totales	15
5. Objetivo	El alumno aplicará diferentes técnicas para el levantamiento y documentación de requerimientos de un sistema.

Temas	Saber	Saber hacer	Ser
1. Técnicas para la obtención de los requerimientos de un sistema.	<ul style="list-style-type: none"> Identificar las principales técnicas para la obtención de requerimientos (Entrevista, Cuestionario, Sesiones JAD). Identificar los pasos para la aplicación de las técnicas para la obtención de requerimientos. 	<ul style="list-style-type: none"> Obtener los requerimientos de un sistema empleando las principales técnicas existentes (Entrevista con usuarios, cuestionario, sesiones JAD). 	<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Ético. Coherente. Discreto. Proactivo. Asertivo. Hábil para comunicarse efectivamente. Líder. Hábil para trabajo en equipo.
2. Especificación de Requerimientos.	<ul style="list-style-type: none"> Identificar los estándares para la especificación de los requerimientos (IEEE, UP). 	<ul style="list-style-type: none"> Elaborar un documento de especificación de requerimientos de acuerdo a un estándar (IEEE, UP). 	<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Coherente. Discreto. Hábil para trabajo en equipo.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

Proceso de evaluación		
Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<ul style="list-style-type: none"> • El alumno elaborará un documento de especificación de requerimientos de un sistema acuerdo a un estándar de especificación. 	<ol style="list-style-type: none"> 1. Identificar el concepto de requerimiento, tipos y característica. 2. Reconocer las técnicas para el levantamiento de requerimientos. 3. Reconocer los estándares para la especificación de requerimientos en un documento. 4. Levantar los requerimientos de un sistema y plasmarlos en un documento de acuerdo a un estándar. 	<ul style="list-style-type: none"> • Ejecución de Tareas. • Listas de verificación. • Desarrollo de Productos.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

Proceso enseñanza aprendizaje	
Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none"> • Ejercicios prácticos. • Estudios de Casos. • Aprendizaje basado en Proyectos. • Instrucción directa. 	<ul style="list-style-type: none"> • Cañón. • Equipo de Cómputo. • Software Ofimática. <li style="background-color: yellow;">• Software de Modelado UML. • Pintarrón. • Plumones.

Espacio Formativo		
Aula	Laboratorio / Taller	Empresa
	X	

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

UNIDADES TEMÁTICAS

1. Unidad Temática	IV. Modelos de Proceso
2. Horas Prácticas	15
3. Horas Teóricas	10
4. Horas Totales	25
5. Objetivo	El alumno conocerá las características, ventajas y desventajas de cada uno de los modelos de proceso para el desarrollo de software.

Temas	Saber	Saber hacer	Ser
1. Proceso Personal de Software PSP.	<ul style="list-style-type: none"> Identificar el objetivo, estructura y características de la metodología PSP. 	<ul style="list-style-type: none"> Aplicar las técnicas de PSP para la recolección de las métricas a nivel PSPO en un proyecto software. 	<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Ético. Coherente. Discreto. Proactivo. Asertivo. Honesto.
2. Proceso de desarrollo de software en equipo.	<ul style="list-style-type: none"> Identificar el objetivo, estructura y características de la metodología TSP. 	<ul style="list-style-type: none"> Integrarse a un equipo de desarrollo de software de acuerdo a los lineamientos establecidos por la metodología TSP. 	<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Ético. Discreto. Proactivo. Asertivo. Dominio personal. Hábil para comunicarse efectivamente. Liderazgo. Hábil para Hábil para trabajo en equipo. Motivador. Honesto.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

3.Proceso Unificado.	<ul style="list-style-type: none"> • Identificar las fases del proceso unificado y sus características. 	<ul style="list-style-type: none"> • Aplicar el proceso unificado en el desarrollo de un sistema. 	<ul style="list-style-type: none"> • Ordenado. • Sistemático. • Objetivo. • Ético. • Coherente.
4.Moprosoft.	<ul style="list-style-type: none"> • Identificar los principales elementos del modelo mexicano MoProsoft. • Identificar su aplicación en la industria de software nacional. 	<ul style="list-style-type: none"> • Reconocer las principales áreas de oportunidad de una organización para la correcta adopción del modelo MoProsoft. 	<ul style="list-style-type: none"> • Ordenado. • Sistemático. • Objetivo. • Ético. • Coherente. • Hábil para trabajo en equipo.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

Proceso de evaluación		
Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<ul style="list-style-type: none"> • El alumno elaborará de un esquema en donde se ubique cada modelo en contexto, con sus principales características, ventajas, desventajas y aplicaciones. • El alumno documentará algunos aspectos de un proyecto software (puede ser de otra asignatura) de acuerdo un modelo de proceso de desarrollo. 	<ol style="list-style-type: none"> 1. Identificar las metodologías y modelos de desarrollo existentes. 2. Identificar las características de la metodología PSP/TSP. 3. Identificar los pasos para la aplicación del Proceso Unificado (UP). 4. Identificar las características de MoProsoft. 	<ul style="list-style-type: none"> • Ejecución de Tareas. • Listas de verificación. • Desarrollo de Productos.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

Proceso enseñanza aprendizaje	
Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none"> • Ejercicios prácticos. • Estudios de Casos. • Aprendizaje basado en Proyectos. • Instrucción directa. 	<ul style="list-style-type: none"> • Cañón. • Equipo de Cómputo. • Software Ofimática. • Software de Modelado UML. • Pintarrón. • Plumones.

Espacio Formativo		
Aula	Laboratorio / Taller	Empresa
	X	

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

UNIDADES TEMÁTICAS

1. Unidad Temática	V. Pruebas y aseguramiento de la calidad
2. Horas Prácticas	10
3. Horas Teóricas	5
4. Horas Totales	15
5. Objetivo	El alumno utilizará técnicas existentes para asegurar la calidad de productos de software.

Temas	Saber	Saber hacer	Ser
1. Calidad en el proceso de Software.	<ul style="list-style-type: none"> Identificar las características de un proceso de software de calidad 		<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Ético. Coherente. Proactivo. Asertivo. Hábil para trabajo en equipo.
2. Calidad en el producto software.	<ul style="list-style-type: none"> Identificar las características principales de un software de calidad 		<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Ético. Coherente. Proactivo. Asertivo. Hábil para trabajo en equipo.
3. Diseño y aplicación de Pruebas.	<ul style="list-style-type: none"> Identificar los tipos y características de las pruebas que se aplican para asegurar la calidad del producto 	<ul style="list-style-type: none"> Diseñar pruebas para la evaluación de la calidad del producto y del proceso de desarrollo de software 	<ul style="list-style-type: none"> Ordenado. Sistemático. Objetivo. Ético. Coherente. Proactivo. Asertivo. Hábil para trabajo en equipo.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

APROBÓ: C. G. U. T.

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

Proceso de evaluación		
Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<ul style="list-style-type: none"> • El alumno elaborará y aplicara un plan de pruebas de un software dado. 	<ol style="list-style-type: none"> 1. Identificar las características de un producto y un proceso de desarrollo de software de calidad. 2. Identificar tipos de pruebas para software. 3. Diseñar pruebas para software. 4. Identificar el proceso de documentación de pruebas software. 	<ul style="list-style-type: none"> • Ejecución de Tareas. • Listas de verificación. • Desarrollo de Productos.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

Proceso enseñanza aprendizaje	
Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none"> • Ejercicios prácticos. • Estudios de Casos. • Aprendizaje basado en Proyectos. • Instrucción directa. 	<ul style="list-style-type: none"> • Cañón. • Equipo de Cómputo. • Software Ofimática. • Software CASE (Testing). • Pintarrón. • Plumones.

Espacio Formativo		
Aula	Laboratorio / Taller	Empresa
	X	

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

CAPACIDADES DERIVADAS DE LAS COMPETENCIAS PROFESIONALES A LAS QUE CONTRIBUYE LA ASIGNATURA

Capacidad	Criterios de Desempeño
<ul style="list-style-type: none">• Estructurar aplicaciones Web avanzadas móviles y de comercio electrónico basado en métodos de ingeniería de software y web, con bases de datos para garantizar la calidad del proceso de desarrollo.	<ul style="list-style-type: none">a) Genera documentos de especificación de requerimientos conforme a los estándares y metodologías establecidas para ello.b) Genera el análisis y modelado de la aplicación de acuerdo a los requerimientos con base en los estándares y metodologías (Patrones de diseño, Ingeniería de Software e Ingeniería Web).c) Genera la aplicación con base en el modelado previamente establecido.d) Ejecuta plan de pruebas para verificar funcionalidad.e) Documenta los resultados.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009

OPTATIVA I: MODELOS DE PROCESO PARA EL DESARROLLO DE SOFTWARE

FUENTES BIBLIOGRÁFICAS

Autor	Año	Título del Documento	Ciudad	País	Editorial
Cem Kane.	1999	<i>Testing Computer Software.</i>	Washington	Estados Unidos de América	Wiley
Craig Larman	2008	<i>Utilización de UML y patrones 2ª. Edición</i>	Nueva Jersey	Estados Unidos de América	Prentice Hall/Pearson
Ian Sommerville, María Isabel Alfonso Galipienso, and Antonio Botia Martínez	2005	<i>Ingeniería de software. 7ª edición</i>	Lancaster	Reino Unido	Pearson Addison Wesley
Perdita Stevens	2003	<i>UML en ingeniería de software con objetos y componentes</i> (http://homepages.inf.ed.ac.uk/perdita/book/)	Londres	Reino Unido	Addison wesley
Ron Patton	2005	<i>Software Testing. 2nd edition</i>	Washington	Estados Unidos de América	Ed. Sams.

ELABORÓ: COMITÉ DE DIRECTORES DE LA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

APROBÓ: C. G. U. T.

REVISÓ: COMISIÓN DE RECTORES PARA LA CONTINUIDAD DE ESTUDIOS

FECHA DE ENTRADA EN VIGOR: SEPTIEMBRE 2009