



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA ACADÉMICA
DIRECCIÓN DE EDUCACIÓN SUPERIOR



PERFIL DOCENTE POR UNIDAD DE APRENDIZAJE

1. DATOS GENERALES

UNIDAD ACADÉMICA: UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS

PROGRAMA ACADÉMICO: Ingeniería Telemática NIVEL I

ÁREA DE FORMACIÓN:	Institucional	Científica Básica	Profesional	Terminal y de Integración
--------------------	---------------	-------------------	-------------	---------------------------

ACADEMIA: Informática UNIDAD DE APRENDIZAJE: Análisis y diseño de sistemas

ESPECIALIDAD Y NIVEL ACADÉMICO REQUERIDO: Licenciatura en computación o afín, de preferencia con maestría o doctorado.

2. OBJETIVO DE LA UNIDAD DE APRENDIZAJE: Implementar en lenguaje C++ sistemas de información, los cuales impliquen su análisis y diseño a través de la Ingeniería del Software (IS) y el Lenguaje de modelado Unificado (UML) para manejar la estructura y flujo de la información para la solución a problemas.

3. PERFIL DOCENTE:

CONOCIMIENTOS	EXPERIENCIA PROFESIONAL	HABILIDADES	ACTITUDES
En programación estructurada y orientada a objetos. En el uso equipo de cómputo como sistema de adquisición de datos. En el usos de protocolos de comunicación. En el diseño de interfaces gráficas. En la programación de microcontroladores.	Docencia. Diseño y desarrollo de instrumentos virtuales.	Manejo de equipo de laboratorio. Interpretación y uso de hojas de especificaciones de componentes electrónicos. Comunicación oral y escrita. Capacidad de Análisis y Síntesis. Manejo de grupos. Manejo de materiales didácticos.	Vocación docente. Honestidad. Ejercicio de la crítica constructiva. Respeto. Tolerancia. Ética. Responsabilidad. Colaboración. Superación docente y profesional. Buena presencia. Compromiso social.

ELABORÓ

Nombre y firma del Presidente de Academia

Ing. Oscar Hernández Fajardo.

REVISÓ

Nombre y firma del Subdirector Académico

AUTORIZO

Nombre del Director de la Unidad Académica

M. en C. Aroldo Rafael Domínguez
DIRECCION



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA ACADÉMICA
DIRECCIÓN DE EDUCACIÓN SUPERIOR



PROGRAMA SINTÉTICO

UNIDAD ACADÉMICA: UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS

PROGRAMA Ingeniería Telemática

ACADÉMICO:

UNIDAD DE APRENDIZAJE: Análisis y Diseño de Sistemas

NIVEL: I

OBJETIVO GENERAL:

Implementar en lenguaje C++ sistemas de información, los cuales impliquen su análisis y diseño a través de la Ingeniería del Software (IS) y el Lenguaje de modelado Unificado (UML) para manejar la estructura y flujo de la información para la solución a problemas.

CONTENIDOS:

- I. Introducción a la Ingeniería de Software
- II. Fundamentos de programación en C++
- III. Fundamentos del paradigma O.O.
- IV. Análisis y Diseño con UML

ORIENTACIÓN DIDÁCTICA:

Desarrollar las habilidades de análisis y abstracción de problemas para el diseño de software a través de prácticas, trabajos, discusiones y conclusiones de conceptos correspondientes a cada unidad temática, así como la demostración de la competencia obtenida en la implementación de un proyecto dirigido a su perfil profesional. El facilitador dará los conceptos para implementar la ingeniería de software y fomentar el análisis de desarrollo del software, deberá proponer prácticas relacionadas para incitar la participación e integración al grupo de trabajo, será responsabilidad indicar los tiempos de revisión para hacer las observaciones y evaluaciones adecuadamente para mejorar su aprendizaje.

EVALUACIÓN Y ACREDITACIÓN:

Para la evaluación y acreditación de la Unidad de Aprendizaje se toma en cuenta la entrega de los trabajos de investigación, tareas, prácticas de laboratorio completas y participación en el aula. Se llevará a cabo la realización de un proyecto que implique todos los temas contenidos. Así como una evaluación exploratoria.

Esta unidad de aprendizaje puede acreditarse también mediante:

Demostración de la competencia para el diseño y desarrollo de un sistema de información mediante el lenguaje C++ y el UML enfocado a su perfil de egreso.

Acreditación en otra Unidad Académica del IPN.

Acreditación en una institución educativa externa al IPN nacional o internacional

BIBLIOGRAFÍA:

Booch, El Lenguaje Unificado De Modelado, 2ª edición, Pearson, Págs: 547, ISBN: 8478290761, ISBN 13: 9788478290765.

Booch, Grady, Análisis de diseño orientado a objetos con aplicaciones, 2ª edición, Encuadernación: Tapa blanda México, ISBN: 9789684443525.

Jacobson, Ivar y Booch, Grady y Rumbaugh, James, El proceso unificado de desarrollo de software, 1ª edición, Addison-Wesley iberoamericana España, 2000, Páginas: 464, ISBN: 9788478290369

Schach, Análisis y Diseño Orientado a Objetos con UML y el Proceso Unific, 1ª edición, Mc-Graw-Hill / Interamericana de México 2005, ISBN: 9789701049822.



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA ACADÉMICA
DIRECCIÓN DE EDUCACIÓN SUPERIOR



UNIDAD ACADÉMICA:

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS.

PROGRAMA ACADÉMICO: Ingeniería Telemática.

PROFESIONAL ASOCIADO: Profesional Asociado en Telemática.

ÁREA FORMATIVA: Científica Básica

MODALIDAD: Presencial

UNIDAD DE APRENDIZAJE: Análisis y diseño de Sistemas

TIPO DE UNIDAD DE APRENDIZAJE:

1. Teórica-Práctica
2. Obligatoria.

VIGENCIA: Enero 2010

NIVEL: I

CRÉDITOS: 9 TEPIC 6.35 SATCA

PROPÓSITO GENERAL

Realizar análisis y diseño con UML para el desarrollo de un sistema de información en Lenguaje C++ y a través de una metodología que permita analizar y estructurar adecuadamente para dar solución a la simplificación y eficiencia de los sistemas de información. Presentando una integración grupal con respeto y responsabilidad hacia la materia, sus compañeros y el medio en donde se desempeñe.

Aportará conocimientos para el fácil entendimiento a la materia de Programación Avanzada, Ingeniería Web, Bases de datos, DSP's, Procesamiento de Imágenes, Estructuras de datos.

OBJETIVO GENERAL

Implementar en lenguaje C++ sistemas de información, los cuales impliquen su análisis y diseño a través de la Ingeniería del Software (IS) y el Lenguaje de modelado Unificado (UML) para manejar la estructura y flujo de la información para la solución a problemas.

TIEMPOS ASIGNADOS

HORAS TEORÍA/SEMANA: 3

HORAS PRÁCTICA/SEMANA: 3

HORAS TEORÍA/SEMESTRE: 54

HORAS PRÁCTICA/SEMESTRE: 54

HORAS TOTALES/SEMESTRE:
108

UNIDAD DE APRENDIZAJE
DISEÑADA Academia de Informática

REVISADA POR: Subdirección Académica

APROBADA POR: S. E. P.
Consejo Técnico Consultivo Escolar

M. en C. Aroldo Rafael Carvallo Domínguez
Presidente del CTCE.

AUTORIZADO POR:
Comisión de Programas Académicos
Consejo General Consultivo del IPN.

SECRETARÍA DE EDUCACIÓN PÚBLICA
INSTITUTO POLITÉCNICO NACIONAL
DIRECCIÓN DE EDUCACIÓN SUPERIOR

Ing. Rodrigo de Jesús Serrano Domínguez
Técnico de la Comisión de Programas Académicos

N° UNIDAD TEMÁTICA: I		NOMBRE: Introducción a la Ingeniería del Software				
COMPETENCIA ESPECÍFICA						
Aplicar la Ingeniería de software sobre sistemas de información donde se requiera análisis y diseño para lograr sistemas reutilizables y con calidad por medio de una metodología.						
No.	CONTENIDOS	HORAS AD Actividades de docencia		HORAS TAA Actividades de Aprendizaje Autónomo		CLAVE BIBLIOGRÁFICA
		T	P	T	P	
1.1	Conceptos fundamentales del software (dato, información).	0.5				1B,2B,4B,5C
1.2	Definición del software.	2.0		0.5		
1.2.1	Características del software.					
1.2.2	Importancia del software.					
1.2.3	Complejidad del diseño y desarrollo del software.					
1.2.4	Crisis del software					
1.2.5	Aspectos a considerar en el desarrollo de software.					
1.2.6	Claves en el diseño del software.					
1.3	Sistemas de información	1.5		0.5		
1.3.1	Definición de los Sistemas de información					
1.3.2	Tipos de sistemas de información					
1.4	Análisis de requisitos	2.5		1.0		
1.4.1	Importancia del Análisis de requisitos.					
1.4.2	Obtención de los requisitos.					
1.4.3	Interpretación y manejo de los requisitos.					
1.5	Ciclos de Vida.	2.0	1.5	1.5		
1.5.1	Cascada.					
1.5.2	Espiral					
1.6	Legislación del software.	0.5				
1.7	Metodologías para el desarrollo del software.	2.5		1.5		
1.7.1	Definición, tipos de metodologías y elementos.					
1.7.2	Análisis de una metodología.					
1.7.3	Desarrollo a pasos de una metodología.					
	Subtotales por Unidad temática:	11.5	1.5	5.0	0.0	
ESTRATEGIAS DE APRENDIZAJE						
Introducción al análisis, requerimientos y diseño del software mediante lluvia de ideas.						
Elaboración de un trabajo de investigación con los temas.						
Elaboración de mapas conceptuales referentes a los temas.						
Desarrollo de prácticas utilizando los ciclos de vida, así como metodologías para el diseño del software.						
Desarrollo de trabajo independiente con relación a los temas propuestos por el profesor.						
Definición de los requerimientos y ciclo de vida del software a diseñar.						
EVALUACIÓN DE LOS APRENDIZAJES						
20% Elaboración de un trabajo el cual deberá contemplar los siguientes elementos: calidad, originalidad, contenido completo (introducción, objetivo, desarrollo, conclusiones y bibliografía).						
10% Elaboración del mapa conceptual el cual deberá de contemplar: calidad, originalidad, contenido completo (que involucre todos los temas vistos) su exposición al grupo.						
30% Selección del modelo de ciclo de vida de acuerdo al proyecto.						
30% Entrega de requerimientos del sistema.						
10% Elaboración de prácticas que contengan: los ejercicios elaborados correctamente: Análisis y requerimientos de sistemas propuestos en clase.						

UNIDAD DE APRENDIZAJE: Análisis y diseño de sistemas

HOJA: 4 DE 11

Nº UNIDAD TEMÁTICA: II		NOMBRE: Fundamentos de Programación en C++.				
COMPETENCIA ESPECÍFICA						
Implementar programas en lenguaje C++ haciendo uso de operadores, sentencias de control y arreglos además de utilizar también el análisis y diseño elaborado en la unidad I para dar solución a sistemas de información.						
No.	CONTENIDOS	HORAS AD Actividades de docencia		HORAS TAA Actividades de Aprendizaje Autónomo		CLAVE BIBLIOGRÁFICA
		T	P	T	P	
2.1	Estructura del lenguaje	0.5				1B, 2B, 4B 12C,
2.2	Funciones de entrada y salida en C++: cin<< y cout>>	0.5				
2.3	Declaración de variables y constantes.					
2.4	Operadores	1.5	1.5	0.5	1.0	
2.4.1	Operadores de asignación.					
2.4.2	Operadores matemáticos: unarios y binarios.					
2.4.3	Operadores Lógicos.					
2.4.4	Operadores Relacionales.					
2.4.5	Operadores Condicionales.					
2.5	Sentencias de Control.					
2.5.1	Sentencias Condicionales: If-else, switch.	1.0	1.5	0.5	1.0	
2.5.2	Sentencias de Repetición: for, while, do-while	1.0	3.0	0.5	1.0	
2.6	Arreglos unidimensionales y bidimensionales.	1.5	3.0	1.0	1.0	
2.6.1	Lectura y Escritura en arreglos.					
2.6.2	Operaciones Básicas con arreglos.					
2.6.3	Métodos de Ordenamiento.					
2.7	Funciones(Métodos)	1.5	3.0	0.5	1.0	
2.7.1	Tipos de funciones.					
2.7.2	Llamado a funciones.					
	Subtotales por Unidad temática:	7.5	12.0	3.0	5.0	
ESTRATEGIAS DE APRENDIZAJE						
Introducción a la estructura de un programa en Lenguaje C++ mediante lluvia de ideas. Realización de ejercicios con sentencias de control utilizando los operadores. Desarrollo de trabajo independiente aplicando las expresiones simples así como las sentencias de control. Solucionar casos prácticos relacionados con operaciones matemáticas utilizando matrices. Solucionar problemas de ordenación con el método de la Burbuja. Elaboración por parte del alumno de un trabajo con los temas vistos. Elaboración de propuesta para el desarrollo de un sistema de información por parte del alumno.						
EVALUACIÓN DE LOS APRENDIZAJES						
20% Elaboración de trabajos relacionados con los temas los cuales deberán contemplar los siguientes elementos: calidad, originalidad, contenido completo (introducción, objetivo, desarrollo, conclusiones y bibliografía). 40% Elaboración de prácticas que contengan: 20% de los ejercicios elaborados correctamente : Código fuente sin errores, Compilados y en ejecución 20% reporte de las prácticas. 20% Elaboración de programas utilizando arreglos, matrices y sentencias de control como trabajo independiente e investigación de conceptos. 20% Evaluación exploratoria.						

N° UNIDAD TEMÁTICA: III		NOMBRE: Fundamentos del Paradigma Orientado a Objetos.				
COMPETENCIA ESPECÍFICA						
Implementar sistemas de información haciendo uso del paradigma Orientado a Objetos en Lenguaje C++ para dar solución a problemas informáticos.						
No.	CONTENIDOS	HORAS AD Actividades de docencia		HORAS TAA Actividades de Aprendizaje Autónomo		CLAVE BIBLIOGRÁFICA
		T	P	T	P	
3.1	Definición del Paradigma Orientado a	0.5				14B,15B
3.2	Objetos.	3.0	4.5	0.5	1.5	
3.2.1	Definición de Clases y Objetos en C++.					
3.2.2	Alcance de Clases en C++.					
3.2.3	Atributos de clases en C++.					
3.3	Definición y creación de Métodos.	3.0	3.0	0.5	1.5	
3.3.1	Constructores en C++.					
3.3.2	Sobrecarga de Métodos en C++.					
3.3.3	Sobrecarga de constructores en C++.					
3.4	Encapsulado de clases	1.5	1.5	0.5	0.5	
3.5	Herencia simple y múltiple.	3.0	2.5		1.5	
3.6	Clases abstractas.	0.5	0.5	0.5	1.0	
3.7	Polimorfismo.	1.0	1.5	0.5	1.5	
	Subtotales por Unidad temática:	12.5	13.5	2.5	7.5	
ESTRATEGIAS DE APRENDIZAJE						
Introducción al Paradigma Orientado a Objetos en C++.						
Realización de ejercicios utilizando clases, objetos, sobrecarga, etc.						
Desarrollo de trabajo independiente aplicando el paradigma orientado.						
Solucionar casos prácticos relacionados con el paradigma orientado a objetos.						
Elaboración por parte del alumno de un trabajo con los temas vistos.						
Elaboración de propuesta para el desarrollo de un sistema de información por parte del alumno utilizando el paradigma orientado a objetos.						
EVALUACIÓN DE LOS APRENDIZAJES						
20% Elaboración de un trabajo el cual deberá contemplar los siguientes elementos: calidad, originalidad, contenido completo (introducción, objetivo, desarrollo, conclusiones y bibliografía).						
40% Elaboración de prácticas que contengan:						
20% de los ejercicios elaborados correctamente						
20% reporte de las prácticas.						
10% Evaluación exploratoria.						
30% Avance del sistema de información tomando requerimientos de la unidad I.						

UNIDAD DE APRENDIZAJE: Análisis y diseño de sistemas

HOJA: 6 DE 11

N° UNIDAD TEMÁTICA: IV		NOMBRE: Análisis y diseño con UML				
COMPETENCIA ESPECÍFICA						
Modelar sistemas de información haciendo uso del lenguaje de modelado unificado para lograr sistemas estandarizados y reutilizables.						
No.	CONTENIDOS	HORAS AD Actividades de docencia		HORAS TAA Actividades de Aprendizaje Autónomo		CLAVE BIBLIOGRÁFICA
		T	P	T	P	
4.1	Introducción a UML.	0.5				3B,6C,8C,9B,11B
4.2	Herramientas del Modelado.	0.5				
4.3	Diagramas básicos en UML.	0.5				
4.3.1	Diagramas de caso de uso.	1.5	1.0	0.5	1.0	
4.3.1.1	Actores					
4.3.1.2	Flujo de la información.					
4.3.1.3	Rol.					
4.4	Diagrama de Actividades.	1.5	1.0	0.5	1.0	
4.4.1	Operaciones					
4.4.2	Transiciones					
4.5	Diagramas de clase.	1.5	1.0	0.5	1.0	
4.5.1	Relaciones					
4.5.2	Atributos.					
4.6	Diagramas de secuencia.	2.0	1.5	1.0	1.5	
4.6.1	Mensajes.					
4.7	Diagrama de Distribución	1.0	0.5	0.5	0.5	
4.8	Generación de código		3.0		1.5	
Subtotales por Unidad temática:		9.0	8.0	3.0	6.5	
ESTRATEGIAS DE APRENDIZAJE						
Introducción al Lenguaje Unificado de Modelado (UML).						
Realización de ejercicios utilizando UML.						
Desarrollo de trabajo independiente aplicando el UML.						
Solucionar casos prácticos relacionados al diseño de un sistema de información utilizando el UML.						
Elaboración por parte del alumno de un trabajo con los temas vistos.						
Modelado y presentación final de un sistema de información por parte del alumno utilizando UML.						
EVALUACIÓN DE LOS APRENDIZAJES						
30% Elaboración de prácticas que contengan los ejercicios elaborados correctamente						
20% Evaluación exploratoria.						
70% Presentación final del análisis, modelado y desarrollo de un sistema de información.						
35% Software desarrollado correctamente: Código fuente sin errores, Compilados y en ejecución						
35% Trabajo escrito desarrollado correctamente: Selección de ciclo de vida, selección de tipo de metodología, análisis y adquisición de requisitos del sistema, definición de clases, objetos y atributos del sistema, elaboración del diagrama respectivo a cada parte del sistema utilizando UML.						



RELACIÓN DE PRÁCTICAS

PRÁCTICA No.	NOMBRE DE LA PRÁCTICA	UNIDADES TEMÁTICAS	DURACIÓN	LUGAR DE REALIZACIÓN
1	Introducción a la ingeniería del software. Objetivo: Definir los requisitos y ciclo de vida en el diseño de software.	I	1.5	LABORATORIO DE COMPUTO
2	Fundamentos de la programación en C++. Objetivo: Desarrollo de programas utilizando las sentencias de control con C++.	II	9.0	
3	Manejo de arreglos unidimensionales y multidimensionales con C++. Objetivo: Desarrollar programas resolver problemas con arreglos bidimensionales y unidimensionales.	II	4.0	
4	Uso de Métodos (funciones) en C++. Objetivo: Desarrollar los programas hechos en las prácticas anteriores aplicando funciones y desarrollo de aplicaciones utilizando diferentes tipos de funciones.	II	4.0	
5	Definición de clases y objetos en C++. Objetivo: Desarrollar programas utilizando clases y objetos en C++.	III	6.0	
6	Manejo de Métodos. Objetivo: Desarrollo de programas utilizando de Métodos en C++.	III	4.5	
7	Herencia simple y múltiple. Objetivo: Desarrollar de programas utilizando herencia simple y múltiple en C++.	III	6.0	
8	Clases Abstractas y polimorfismo. Objetivo: Desarrollar programas donde se aplique las clases abstractas y polimorfismo.	III	4.5	
9	Diagramas de casos de uso. Objetivo: Desarrollar los casos de uso a partir de los requerimientos de la unidad I	IV	2.0	



UNIDAD DE APRENDIZAJE: Análisis y diseño de sistemas

HOJA: 7 DE 11

PRÁCTICA No.	NOMBRE DE LA PRÁCTICA	UNIDADES TEMÁTICAS	DURACIÓN	LUGAR DE REALIZACIÓN
10	Diagrama de actividades Objetivo: Elaborar los diagramas de actividades a partir de los diagramas de casos de uso.	IV	2.0	LABORATORIO DE COMPUTO
11	Diagramas de Clase. Objetivo: Diseñar el diagrama de clases retomando los requerimientos de la unidad I y los diagramas anteriores.	IV	2.0	
12	Diagramas de Secuencia Objetivo: Desarrollar los diagramas de secuencia en base a diagrama de clases diseñado.	IV	3.0	
13	Diagramas de Distribución. Objetivo: Desarrollar diagramas de distribución en base a la arquitectura del sistema.	IV	1.0	
14	Generación de código. Objetivo: Generación de código a partir de la herramienta de diseño.	IV	4.5	
		TOTAL DE HORAS	54	

EVALUACIÓN Y ACREDITACIÓN:

Para cada una de las unidades temáticas se considera de un 20% al 30% del 100% del total de evaluación. La parte práctica de esta unidad de aprendizaje será evaluada considerando la asistencia al laboratorio de Cómputo, el desarrollo del código fuente junto con su Diagrama de Flujo para cada programa, entrega de los programas compilados, funcionando correctamente y la elaboración del correspondiente reporte escrito con su contenido completo: Objetivo, Metodología, Código, Diagrama de Flujo y corrida. Así como el desarrollo correcto de los diagramas en UML. Será indispensable presentar todas las prácticas y reportes escritos realizados para tener derecho de acreditar la unidad de aprendizaje y presentar el Examen Extraordinario.



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA ACADÉMICA
DIRECCIÓN DE EDUCACIÓN SUPERIOR



UNIDAD DE APRENDIZAJE:

Análisis y diseño de sistemas

HOJA: 9 DE 11

PROCEDIMIENTO DE EVALUACIÓN

Haber asistido al menos el 80% de las clases teórico-prácticas impartidas para tener derecho a ser evaluado.

1. Valor de las unidades temáticas dentro de la unidad de aprendizaje:

- Unidad temática I 25%
- Unidad temática II 25%
- Unidad temática III 25%
- Unidad temática IV 25%

Esta unidad de aprendizaje puede acreditarse también mediante:

· Demostración de competencia para el desarrollo de programas aplicados a la ingeniería telemática, a más tardar tres semanas después de iniciado el curso. Presentando las evidencias de las evaluaciones correspondientes a cada unidad temática

La Subdirección Académica en conjunto con la Academia determinará la equivalencia con Unidades de Aprendizajes de otras Unidades Académicas tanto del IPN como externas.

Para la evaluación de la unidad de aprendizaje:

- Desarrollo de un proyecto;
- Programas de cómputo;
- Reportes escritos, y
- Ensayos.

Para acreditar la unidad de aprendizaje por "competencia demostrada":

- Realizar dos análisis y modelados de aplicaciones, en donde se evalúe lo expuesto en las unidades temáticas I, II, III y IV, así como el desarrollo de un caso práctico, en conjunto con la Academia.



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA ACADÉMICA
DIRECCIÓN DE EDUCACIÓN SUPERIOR



UNIDAD DE APRENDIZAJE:

Análisis y diseño de sistemas

HOJA: 10

DE 11

CLAVE	B	C	BIBLIOGRAFÍA
1.	X		Kendall & Kendall, "Análisis y Diseño de Sistemas", ed. Pearson Prentice Hall.
2.	X		Roger Pressman, "Ingeniería de Software", ed. McGraw Hill, Quinta edición, 2002
3.	X		Alfredo Weitzenfeld, "Ingeniería de Software Orientada a Objetos con UML. Java e internet", ed. Thomson.
4.	X		D. Marca, C. McGowan, "Structured Analysis and Design Technique", McGraw-Hill, 1987, ISBN 0-07-040235-3.
5.		X	Ian Sommerville, "Ingeniería de Software", Addison Wesley
6.		X	Craig Larman, "UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado", ed. Prentice Hall.
7.		X	Grady Booch; Ivar Jacobson; James Rumbaugh; Jesús J. García Molina, " <u>El Lenguaje Unificado De Modelado: Guía Del Usuario</u> ", Ed. Addison Wesley, ISBN: 8478290761, 2da edición, 2006, pag. 552
8.		X	Martin Fowler, Kendall Scott, "UML Gota a Gota", ed. Addison Wesley, 1999
9.	X		Debrauwer, Heyde, Fien Van Der, "UML 2: Iniciación, Ejemplos Y Ejercicios Corregidos (RECURSOS INFORMATICOS)", Primera Edición, 2005
10.	X		Grady Booch; Ivar Jacobson; James Rumbaugh; Jesús J. García Molina, "El Lenguaje Unificado De Modelado: Guía Del Usuario", Ed. Addison Wesley, Segunda edición, 2006
11.	X		Neustadt, Ila; Arlow, Jim, "Uml 2", Ed. Anaya, primera edición, 2006
12.		X	Martin Fowler, "La Nueva Metodología", [en línea] http://www.programacionextrema.org/articulos/newMethodology.es.html , revisado el 31 de mayo de 2009.
13.		X	Robert Glass, "Software Creativity", Prentice-Hall, 1995
14.	X		Deitel, "C++ cómo programar", ed. Pearson Prentice Hall, cuarta edición, 2003
15.	X		Luis Joyanes Aguilar, "Programación en C++: Algoritmos, Estructuras de Datos Y Objetos", McGraw-Hill Interamericana, 2006