

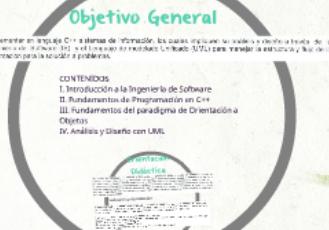
Unidad de Aprendizaje: Análisis y Diseño de Sistemas

Docente:
Ing. Lamberto Maza Casas
Enero, 2020.



Unidad de Aprendizaje: Análisis y Diseño de Sistemas

Docente:
Ing. Lamberto Maza Casas
Enero, 2020.



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 de Orientación a Objetos
- 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 y desarrollo del software, deberá proponer prácticas relacionadas para iniciar 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:
Demuestra 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

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

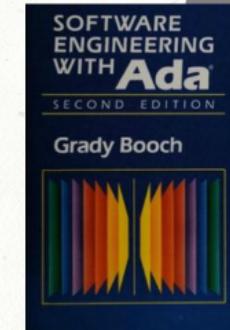
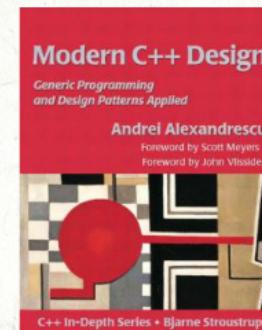
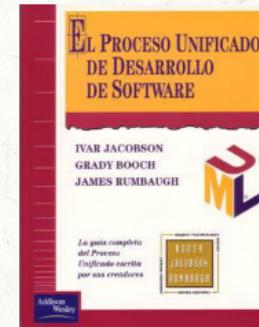
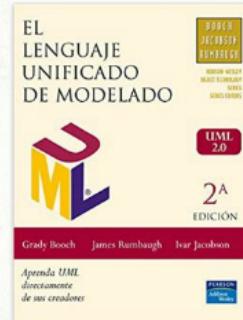
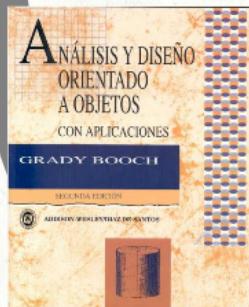
Booch, El Lenguaje Unificado De Modelado, 2^a edición, Pearson, Págs: 547, ISBN: 8478290761, ISBN 13: 9788478290765.

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

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

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

Portadas de algunos de los libros de la bibliografía et al



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 solucionar 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.

Tiempos Asignados:

TIEMPOS ASIGNADOS

HORAS TEORÍA/SEMANA: 3

HORAS PRÁCTICA/SEMANA: 3

HORAS TEORÍA/SEMESTRE: 54

HORAS PRÁCTICA/SEMESTRE: 54

HORAS TOTALES/SEMESTRE:
108

U.T. I 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.

U.T. II 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.

U.T. I 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.

U:
P:
Com
leng
de c
anál
solu

U.T. II Fundamentos de Programación en C++

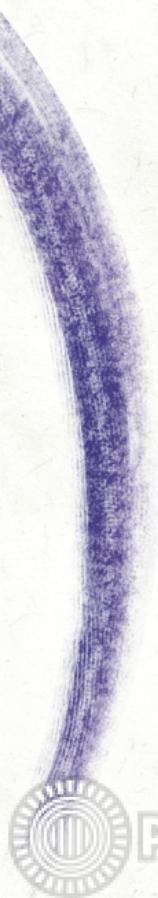
Competencia específica: Implementar programas en lenguaje C++ haciendo uso de operadores, sentencias de control y arreglos, además de utilizar también el análisis y diseño elaborado en la unidad I para dar solución a sistemas de información.

U.T. III Fundamentos del paradigma de Orientación a Objetos

Competencia específica: Implementar sistemas de información haciendo uso del paradigma de orientación a objetos en lenguaje C++ para dar solución a problemas informáticos.

U.T. IV Análisis y diseño con UML

Competencia específica: Modelar sistemas de información haciendo uso del lenguaje de modelado unificado para lograr sistemas estandarizados y reutilizables.



U.T. I Introducción a la Ingeniería del Software

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				
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	Ánalysis de una metodología.					
1.7.3	Desarrollo a pasos de una metodología.					
Subtotales por Unidad temática:		11.5	1.5	5.0	0.0	

U.T. II Fundamentos de Programación en C++

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				
2.2	Funciones de entrada y salida en C++: cin<< y cout>>	0.5				
2.3	Declaración de variables y constantes.					
2.4	Operadores					
2.4.1	Operadores de asignación.	1.5	1.5	0.5	1.0	
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	

U.T. III Fundamentos del Paradigma de Orientación a Objetos

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 Objetos.	0.5				
3.2		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			
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	

U.T. IV Análisis y Diseño con UML

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				
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

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	

Relación de Prácticas (Cont. ...)

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		

PRÁCTICA No.	NOMBRE DE LA PRÁCTICA	UNIDADES TEMÁTICAS	DURACIÓN	LUGAR DE REALIZACIÓN	DE
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

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.

¡Éxito a todos!

Unidad de Aprendizaje: Análisis y Diseño de Sistemas

Docente:
Ing. Lamberto Maza Casas
Enero, 2020.

