



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA ACADÉMICA

DIRECCIÓN DE EDUCACIÓN SUPERIOR

PROGRAMA SINTÉTICO

UNIDAD ACADÉMICA: ESCUELA SUPERIOR DE CÓMPUTO

PROGRAMA Ingeniería en Sistemas Computacionales

ACADÉMICO:

UNIDAD DE APRENDIZAJE: Programación Orientada a Objetos

NIVEL: II

OBJETIVO GENERAL:

Crear aplicaciones de software, para la solución de problemas computacionales, mediante el uso del paradigma orientado a objetos y la utilización de un lenguaje de programación.

CONTENIDOS:

- I. Fundamentos.
- II. Clases y objetos.
- III. Relaciones entre objetos.
- IV. Polimorfismo

ORIENTACIÓN DIDÁCTICA:

Las actividades que se llevarán a cabo contribuirán a desarrollar habilidades de abstracción, indagación de información y solución de problemas mediante la realización de una aplicación de software, la creación de programas de cómputo simples, exposición y discusiones dirigidas. Dichas actividades, fomentarán el trabajo colaborativo. La aplicación de software, se construirá a lo largo del curso.

El docente decidirá cuáles serán las características que debe cumplir la aplicación de software en cada entrega. Adicionalmente fijará los tiempos de realización y de entrega para cada actividad. Finalmente, llevará un registro de los exámenes y los programas de cómputo realizados durante el curso.

EVALUACIÓN Y ACREDITACIÓN:

- Registro de productos elaborados en clase (programas de cómputo, ejercicios).
- Exámenes.
- Proyecto: aplicación de software empleando el paradigma orientado a objetos, evaluado en cada periodo.

Esta unidad de aprendizaje también puede acreditarse mediante:

- Realización de examen teórico-práctico para demostrar competencia para programación orientada a objetos.
- Acreditación en otra UA del IPN.
- Acreditación en una institución educativa externa al IPN nacional o internacional

BIBLIOGRAFÍA:

Booch Grady. Análisis y diseño orientado a objetos, con aplicaciones. 2^{da} ed., Pearson Educación. Santa Clara, California, 1996, págs. 27-162, ISBN: 9684443528.

Budd Timothy. An introduction to Object Oriented Programming. 3^{ra} ed., Addison Wesley, 2001, 648 págs., ISBN: 0201760312.

Muñoz Caro Camelia, Niño Ramos Alfonso, Vizcaino Barceló Aurora. Introducción a la programación con orientación a objetos. Pearson Educación. Madrid, 2002. 408 págs., ISBN: 8420534404.

O'Docherty Mike. Object-Oriented Analysis and Design. Understanding System Development with UML 2.0. John Wiley & Sons. England. págs 9-97, ISBN: 0-470-09240-8.



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA ACADÉMICA

DIRECCIÓN DE EDUCACIÓN SUPERIOR

UNIDAD ACADÉMICA: ESCUELA SUPERIOR DE CÓMPUTO
PROGRAMA ACADÉMICO: Ingeniería en Sistemas Computacionales.
PROFESIONAL ASOCIADO: Programador Analista.
ÁREA FORMATIVA: Profesional
MODALIDAD: Presencial

UNIDAD DE APRENDIZAJE: Programación Orientada a Objetos.
TIPO DE UNIDAD DE APRENDIZAJE: 1) Teórico-práctica/Obligatoria
VIGENCIA: 2009
NIVEL: II
CRÉDITOS: 7.5 TEPIC – 4.34 SATCAA

PROPÓSITO GENERAL

Proporcionar una herramienta fundamental para solucionar problemas en distintos ámbitos, a través del diseño de sistemas computacionales, los cuales deben funcionar eficientemente.

Competencias para la unidad de aprendizaje:

- Conocer la filosofía del paradigma orientado a objetos, para comprender su importancia en el desarrollo de sistemas computacionales.
- Dominar los conceptos de abstracción, encapsulamiento, herencia y polimorfismo, para proponer soluciones a problemas computacionales, a través de la creación de programas simples en un lenguaje de programación.
- Diferenciar entre los conceptos de sobrecarga y sobreescritura, para utilizarlos adecuadamente, en el diseño de clases.
- Propiciar el aprendizaje autónomo, para adquirir nuevos conocimientos, a través de la indagación de conceptos del paradigma orientado a objetos.
- Desarrollar habilidad para trabajar en equipo, al construir aplicaciones de software.

Relaciones con las competencias de otras unidades de aprendizaje:

- Vertical
 - Ingeniería de Software
- Horizontal
 - Análisis y diseño orientado a objetos
 - Tecnologías para la web

OBJETIVO GENERAL

Crear aplicaciones de software, para la solución de problemas computacionales, mediante el uso del paradigma orientado a objetos y la utilización de un lenguaje de programación.

TIEMPOS ASIGNADOS

HORAS TEORÍA/SEMANA: 3

HORAS PRÁCTICA/SEMANA: 1.5

HORAS TEORÍA/SEMESTRE: 54

HORAS PRÁCTICA/SEMESTRE: 27

HORAS TOTALES/SEMESTRE: 81

UNIDAD DE APRENDIZAJE REDISEÑADA POR: Academia de Algoritmia y Programación.

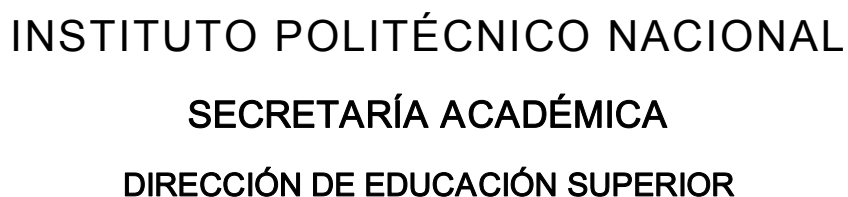
REVISADA POR: Subdirección Académica.

APROBADA POR:
Consejo Técnico Consultivo Escolar.
2009
Ing. Apolinar Francisco Cruz Lázaro
Presidente del CTCE.
Sello de la UA

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

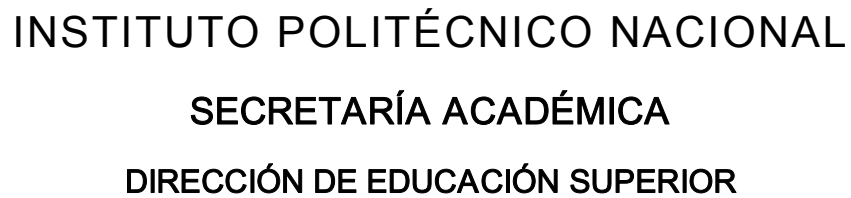
2009

Dr. David Jaramillo Viguera
Secretario Técnico de la Comisión de Programas Académicos
Sello Oficial de la DES



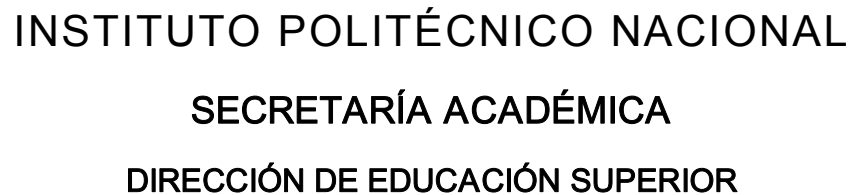
HOJA: 3 DE 9

N° UNIDAD TEMÁTICA: I		NOMBRE: Fundamentos.				
OBJETIVO PARTICULAR						
Conocer la filosofía del paradigma orientado a objetos, para tener una visión general de su aplicación, en el desarrollo de sistemas computacionales eficientes.						
No.	CONTENIDOS	HORAS AD Actividades de docencia (a)		HORAS TAA Actividades de Aprendizaje Autónomo (b)		CLAVE BIBLIOGRÁFICA
		T	P	T	P	
1.1 1.2 1.2.1 1.2.2 1.2.3 1.2.4	Antecedentes. Modelo orientado a objetos. Principio de abstracción. Principio de modularidad. Principio de jerarquía. Principio de tipificación.	0.5 0.5	0.5	1 1	1	1C, 4C
Subtotales por Unidad temática:		1	0.5	2	1	
ESTRATEGIAS DE APRENDIZAJE						
<ul style="list-style-type: none">● Investigación previa sobre los temas de la unidad temática.● Discusión dirigida de los conceptos.● Elaboración de mapa conceptual en pequeños grupos, sobre los principios del paradigma orientado a objetos.● Elaboración de ejemplos en los que se apliquen los conceptos de la unidad temática.● Resolución de problemas.						
EVALUACIÓN DE LOS APRENDIZAJES						
20% Mapas conceptuales, los cuales deberá describir los conceptos de la unidad temática.						
80% Resolución de problemas, en los que se evaluará la correcta aplicación de los conceptos del paradigma orientado a objetos.						



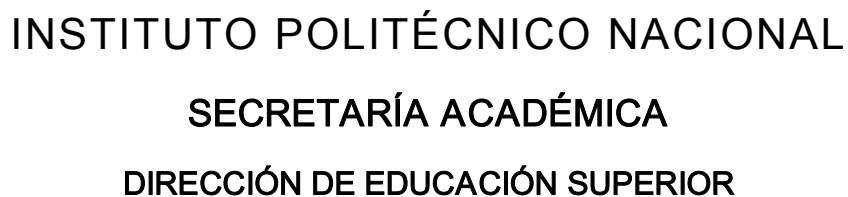
HOJA: 4 DE 9

N° UNIDAD TEMÁTICA: II		OBJETIVO PARTICULAR				NOMBRE: Clases y objetos	
Crear programas de cómputo simples, para solucionar problemas sencillos, utilizando los conceptos de abstracción, encapsulamiento y sobrecarga.							
No.	CONTENIDOS	HORAS AD Actividades de docencia (a)		HORAS TAA Actividades de Aprendizaje Autónomo (b)		CLAVE BIBLIOGRÁFICA	
		T	P	T	P		
2.1 2.1.1 2.1.2 2.1.3	Clases y objetos Atributos y métodos Constructores y destructores Variables y métodos de clase.	5	2	7	4	2B, 5C	
2.2 2.2.1 2.2.2	Sobrecarga Sobrecarga de funciones Sobrecarga de operadores	3	1.5	6	3		
Subtotales por Unidad temática:		8	3.5	13	7		
ESTRATEGIAS DE APRENDIZAJE							
<ul style="list-style-type: none">● Indagación previa sobre los conceptos de la unidad temática.● Elaboración de programas de cómputo simples, donde se apliquen los conceptos revisados en la unidad temática.● Diseño e implementación de clases, para su proyecto, a través del trabajo en equipo● Resolución de examen.							
EVALUACIÓN DE LOS APRENDIZAJES							
30% Programas de cómputo (Relación de prácticas). Deberán ejecutarse correctamente y no deberán tener errores. 20% Diseño de clases del proyecto, utilizando un diagrama de clases, especificando atributos y métodos para cada clase. 50% Examen.							



HOJA: 5 DE 9

N° UNIDAD TEMÁTICA: III		NOMBRE: Relaciones entre objetos				
OBJETIVO PARTICULAR						
Diseñar una jerarquía de clases adecuada, para reutilizar código ya existente, utilizando las distintas relaciones entre objetos.						
No.	CONTENIDOS	HORAS AD Actividades de docencia (a)		HORAS TAA Actividades de Aprendizaje Autónomo		CLAVE BIBLIOGRÁFICA
		T	P	T	P	
3.1	Agregación	1.5	0.5	4.5	2.5	2B, 3B, 4C
3.2	Herencia	3	1	6	3.5	
3.2.1	Herencia simple					
3.2.2	Herencia múltiple					
Subtotales por Unidad temática:		4.5	1.5	10.5	6	
ESTRATEGIAS DE APRENDIZAJE						
<ul style="list-style-type: none">● Indagación previa de los conceptos● Lluvia de ideas para crear la jeraquía de clases en la solución de diversos problemas.● Elaboración de programas de cómputo, que involucren la creación de una jeraquía de clases.● Elaboración e implementación de la jeraquía de clases de su proyecto.● Resolución de examen.						
EVALUACIÓN DE LOS APRENDIZAJES						
20% Programas de cómputo (Relación de prácticas). Deberán ejecutarse correctamente y no deberán tener errores.						
40% Diseño de clases del proyecto, especificando las relaciones entre ellas, utilizando un diagrama UML.						
40% Examen.						



HOJA: 6 DE 9

N° UNIDAD TEMÁTICA: IV		NOMBRE: Polimorfismo				
OBJETIVO PARTICULAR						
Construir una aplicación de software, para diseñar un sistema computacional eficiente, a través del uso de polimorfismo.						
No.	CONTENIDOS	HORAS AD Actividades de docencia (a)		HORAS TAA Actividades de Aprendizaje Autónomo (b)		CLAVE BIBLIOGRÁFICA
		T	P	T	P	
4.1	Enlace estático y enlace dinámico	0.5	0.5	2.5	1	2B,3B,4C
4.2	Clases abstractas	1	0.5	5	2.5	
4.3	Forzamiento	1	0.5	5	2.5	
4.3.1	Forzamiento hacia arriba (upcasting)					
4.3.2	Forzamiento hacia abajo (downcasting)					
Subtotales por Unidad temática:		2.5	1.5	12.5	6	
ESTRATEGIAS DE APRENDIZAJE						
<ul style="list-style-type: none">● Investigación previa sobre los temas que se abordarán en clase.● Resolución de problemas a través de trabajo en equipo.● Elaboración de programas de cómputo, aplicando polimorfismo.● Exposición ante el grupo, para mostrar el diseño y funcionamiento de su proyecto.						
EVALUACIÓN DE LOS APRENDIZAJES						
20% Programas de cómputo (Relación de prácticas). Deberán ejecutarse correctamente y no deberán tener errores.						
40% Aplicación de software, la cual deberá apegarse al diagrama de clases previamente diseñado y deberá funcionar correctamente.						
10% Manual técnico y manual de usuario de la aplicación desarrollada. Ambos manuales deberán tener una redacción adecuada. El manual técnico deberá incluir diagrama de casos de uso y diagrama de clases. El manual de usuario, deberá describir adecuadamente cómo instalar y operar la aplicación de software.						
10% Exposición del proyecto. Para evaluarla se considerará la elaboración de diapositivas y la exposición verbal ante el grupo.						



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA ACADÉMICA

DIRECCIÓN DE EDUCACIÓN SUPERIOR

UNIDAD DE APRENDIZAJE: Programación Orientada a Objetos

HOJA: 7 DE 9

RELACIÓN DE PRÁCTICAS

PRÁCTICA No.	NOMBRE DE LA PRÁCTICA	UNIDADES TEMÁTICAS	DURACIÓN	LUGAR DE REALIZACIÓN
1	¿Cómo compilar programas orientados a objetos?	I	1.5	Laboratorios de la unidad académica
2	Clases e instancias de clase.	II	4.5	
3	Uso de variables y métodos de clase.	II	1.5	
4	Sobrecarga de funciones.	II	1.5	
5	Aplicaciones de la sobrecarga de operadores.	II	3	
6	Uso de la agregación.	III	3	
7	Jerarquía de clases.	III	4.5	
8	Clases abstractas.	IV	4.5	
9	Forzamientos.	IV	3	
		TOTAL DE HORAS	27	

EVALUACIÓN Y ACREDITACIÓN:

Para evaluar las prácticas se considerarán los siguientes aspectos:

- Los programas de cómputo deberán ejecutarse correctamente y no deberán tener errores.
- En cada programa de cómputo se debe aplicar el paradigma orientado a objetos.
- Se cuidará el estilo de programación.

Las prácticas se evalúan según está indicado en cada unidad temática. Siendo indispensable aprobarlas para poder acreditar la unidad de aprendizaje.



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA ACADÉMICA

DIRECCIÓN DE EDUCACIÓN SUPERIOR

UNIDAD DE APRENDIZAJE: Programación Orientada a Objetos

HOJA: 8 DE 9

PROCEDIMIENTO DE EVALUACIÓN

Los procedimientos para evaluar y acreditar la unidad de aprendizaje de Programación Orientada a Objetos es la siguiente:

Unidad Temática I	10%
Unidad Temática II	30%
Unidad Temática III	20%
Unidad Temática IV	20%

Esta unidad de aprendizaje también puede acreditarse mediante:

- La demostración de la competencia en tres semanas, a partir del inicio del curso. Para lo cual, se deberán realizar los trabajos y proyectos marcados como evaluación en cada una de las unidades temáticas.
- Acreditación en otra UA del IPN, previa aprobación de la Academia
- Acreditación en una institución educativa externa al IPN nacional o internacional, previa aprobación de la Academia

CLAVE	B	C	BIBLIOGRAFÍA
1		X	Booch Grady. <u>Análisis y diseño orientado a objetos, con aplicaciones</u> 2 ^{da} ed., Pearson Educación. Santa Clara, California, 1996, págs. 27-162, ISBN: 9684443528.
2	X		Budd Timothy. <u>An introduction to Object Oriented Programming.</u> 3 ^{ra} ed., Addison Wesley, 2001, 648 págs., ISBN: 0201760312.
3	X		Muñoz Caro Camelia, Niño Ramos Alfonso, Vizcaíno Barceló Aurora. <u>Introducción a la programación con orientación a objetos.</u> Pearson Educación. Madrid, 2002. 408 págs., ISBN: 8420534404.
4		X	O'Docherty Mike. <u>Object-Oriented Analysis and Design. Understanding System Development with UML 2.0.</u> John Wiley & Sons. England. págs 9-97, ISBN: 0-470-09240-8
5		X	Stroustrup Bjarne. <u>What is Object Oriented Programming?</u> , Proc. 1 st European Software Festival, February 1991.



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA ACADÉMICA

DIRECCIÓN DE EDUCACIÓN SUPERIOR

PERFIL DOCENTE POR UNIDAD DE APRENDIZAJE

1. DATOS GENERALES

UNIDAD ACADÉMICA: ESCUELA SUPERIOR DE CÓMPUTO

PROGRAMA ACADÉMICO: Ingeniería en Sistemas Computacionales NIVEL II

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

ACADEMIA: Algoritmia y programación.

UNIDAD DE APRENDIZAJE: Programación orientada a objetos.

ESPECIALIDAD Y NIVEL ACADÉMICO REQUERIDO: Ingeniería en Sistemas ó licenciatura en Computación

2. **OBJETIVO DE LA UNIDAD DE APRENDIZAJE:** Crear aplicaciones de software, para la solución de problemas computacionales, mediante el uso del paradigma orientado a objetos y la utilización de un lenguaje de programación.

3. PERFIL DOCENTE:

CONOCIMIENTOS	EXPERIENCIA PROFESIONAL	HABILIDADES	ACTITUDES
<ul style="list-style-type: none">● Paradigma orientado a objetos.● Lenguajes de programación orientado a objetos.● Conocer el MEI.	<ul style="list-style-type: none">● Experiencia diseñando sistemas computacionales● Experiencia programando usando objetos.● Experiencia en manejo de grupos y trabajo colaborativo.	<ul style="list-style-type: none">● Capacidad para expresarse correctamente de forma oral y escrita.● Capacidad para revisar código de programas y hallar errores.● Facilidad para el manejo de grupos.● Aplicar el proceso educativo del MEI.	<ul style="list-style-type: none">● Respeto hacia los otros.● Paciencia al explicar a otros.● Sencillez al tratar con otras personas.● Creatividad.● Flexibilidad para abordar problemas desde distintos puntos de vista.

ELABORÓ

M. en C. Ericka Hernández Rubio

REVISÓ

M. en C. Flavio Arturo Sánchez Garfias

AUTORIZÓ

Ing. Apolinar Francisco Cruz Lázaro