


3.7 ÁREA DE PROFUNDIZACIÓN II: CIENCIAS DE LA COMPUTACIÓN Y MATEMÁTICAS

		UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO Escuela Nacional de Estudios Superiores Unidad Juriquilla		 ENES JURIQUILLA		
		Plan de Estudios de la Licenciatura en Tecnología Sistema Escolarizado: Modalidad Presencial				
Programa de estudios de la asignatura Programación Orientada a Objetos						
Clave	Semestre 3	Créditos 8	Duración	16 semanas		
			Eje de formación	Terminal		
			Área de profundización	Ciencias de la Computación y Matemáticas		
			Etapas de formación	Básica		
Modalidad	Curso (X) Taller (X) Lab () Sem ()		Tipo	T () P () T/P (X)		
Carácter	Obligatorio () Optativo () Obligatorio E (X) Optativo E ()		Horas			
			Semana		Semestre	
			Teóricas	2	Teóricas	32
			Prácticas	4	Prácticas	64
			Total	6	Total	96
Seriación						
Ninguna ()						
Obligatoria (X)						
Asignatura antecedente			Fundamentos de Programación (solo para el área de profundización en Ciencias de la Computación y Matemáticas)			
Asignatura subsecuente						
Indicativa ()						
Asignatura antecedente						
Asignatura subsecuente						

Objetivos generales:

Al finalizar el curso el alumnado será capaz de construir y evaluar algoritmos computacionales utilizando el paradigma de programación orientada a objetos para diseñar e implementar software modular y reutilizable.

Objetivos específicos:

El alumnado comprenderá los conceptos fundamentales de la programación orientada a objetos.

El alumnado conocerá las vistas en el diseño orientado a objetos para aplicarlo en la solución de problemas.

El alumnado aplicará las técnicas y herramientas del paradigma de programación orientada a objetos.

El alumnado aplicará las distintas propiedades de la programación orientada a objetos para la resolución de problemas.

El alumnado comprenderá los distintos tipos de errores y excepciones para generar programas y aplicaciones validadas.

El alumnado construirá aplicaciones con el principio de flujo de entrada y salidas para procesar información partiendo de un problema.

El alumnado aplicará conceptos avanzados de la programación orientada objetos para solución de problemas complejos.

Índice temático

	Tema	Horas Semestre	
		Teóricas	Prácticas
1	Introducción a la programación orientada a objetos (POO)	2	8
2	UML	2	6
3	Tipos, expresiones y control de flujo	8	12
4	Herencia y polimorfismo	8	12
5	Manejo de excepciones y errores	4	10
6	Flujo de entrada y salida	4	8
7	Programación de hilos	4	8
Subtotal		32	64
Total		96	

Contenido Temático

Tema	Subtemas
1	Introducción a la programación orientada a objetos (POO) 1.1 Elementos básicos del paradigma de POO 1.1.1 Tipos de datos primitivos y abstractos 1.1.2 Objetos 1.2 Propiedades básicas del paradigma de POO 1.2.1 Abstracción 1.2.2 Cohesión 1.2.3 Encapsulamiento 1.2.4 Modularidad

	<ul style="list-style-type: none"> 1.2.5 Herencia 1.2.6 Polimorfismo 1.2.7 Acoplamiento 1.2.8 Jerarquía de clases
2	<p>UML</p> <ul style="list-style-type: none"> 2.1 Diseño estático 2.2 Diseño dinámico
3	<p>Tipos, expresiones y control de flujo</p> <ul style="list-style-type: none"> 3.1 Generalidades <ul style="list-style-type: none"> 3.1.1 Identificadores 3.1.2 Palabras reservadas 3.1.3 Comentarios 3.1.4 Clases 3.1.5 Objetos 3.2 Tipos de datos <ul style="list-style-type: none"> 3.2.1 Primitivos y jerarquías 3.2.2 Referencias 3.2.3 Conversiones de tipos 3.2.4 Operadores aritméticos 3.2.5 Operadores de asignación 3.2.6 Operadores relacionales 3.2.7 Operadores especiales 3.2.8 Operadores a nivel de bit 3.2.9 Operadores lógicos 3.3 Arreglos 3.4 Tipos y ámbito de las variables <ul style="list-style-type: none"> 3.4.1 Variables estáticas 3.4.2 Variables constantes 3.5 Tipos de clases <ul style="list-style-type: none"> 3.5.1 Públicas 3.5.2 Sin modificador 3.5.3 Abstractas 3.5.4 Finales 3.5.5 Internas
4	<p>Herencia y Polimorfismo</p> <ul style="list-style-type: none"> 4.1 Herencia 4.2 Constructores 4.3 Polimorfismo 4.4 Referencias (this y base) 4.5 Modificadores de acceso (encapsulamiento) 4.6 Interfaces



	4.7 Plantilas	
5	Manejo de excepciones y errores 5.1 Definición de error y excepción 5.2 Jerarquía de clases 5.3 Estructura try-catch-finally 5.4 Manejo de errores y excepciones	
6	Flujo de entrada y salida 6.1 Archivos de entrada y salida 6.2 Jerarquía de clases de los flujos de datos 6.3 Manipulación de archivos y directorios 6.4 Flujos de entrada de datos 6.4.1 Lectura de archivos 6.4.2 Lectura de teclado 6.5 Flujo de salida de datos (archivos)	
7	Programación de hilos Definición de un hilo Ciclo de vida de un hilo Control básico Clases para el uso de hilos Planificador y prioridades Métodos de sincronización	
Estrategias didácticas		Evaluación del aprendizaje
Exposición	(X)	Exámenes parciales (X)
Trabajo en equipo	(X)	Examen final (X)
Lecturas	(X)	Trabajos y tareas (X)
Trabajo de investigación	(X)	Presentación de tema ()
Prácticas (taller o laboratorio)	()	Participación en clases (X)
Prácticas de campo	()	Asistencia (X)
Aprendizaje por proyectos	()	Rúbricas ()
Aprendizaje basado en problemas	(X)	Portafolios ()
Casos de enseñanza	(X)	Listas de cotejo ()
Otras (especificar)	()	Otras (especificar) ()
Código de conducta		
<p>La conducta del profesorado y alumnado del curso será acorde con los principios y valores especificados en el Código de Ética de la Universidad Nacional Autónoma de México aprobado el 1 de julio del 2015 por el Consejo Universitario, en especial en lo referente a la integridad y honestidad académica. “La integridad y la honestidad académica implican: Citar las fuentes de ideas, textos, imágenes, gráficos u obras artísticas que se empleen en el trabajo universitario, y no sustraer o tomar la información generada por otros o por sí mismo sin señalar la cita correspondiente u obtener su consentimiento y acuerdo. No falsificar, alterar, manipular, fabricar, inventar o fingir la autenticidad de datos, resultados, imágenes o información en los trabajos académicos, proyectos de investigación, exámenes, ensayos, informes, reportes, tesis, audiencias, procedimientos de orden</p>		



disciplinario o en cualquier documento inherente a la vida académica universitaria” (Gaceta UNAM, 30 de julio 2015).

Perfil Profesiográfico	
Título o Grado	Deberá contar con licenciatura o posgrado en ciencias o ingeniería de la computación o bien alguna otra afín del área de las Ciencias Físico Matemáticas y las Ingenierías.
Experiencia docente	Con experiencia docente en licenciatura y/o en posgrado, preferentemente de tres años impartiendo la asignatura u otra relacionada en el nivel superior.
Otra característica	El curso-taller deberá ser impartido por académicas/os de tiempo completo de la UNAM, o equivalente, que estén activos en investigación sobre temas directamente relacionados con el curso-taller.
Bibliografía básica	
<ul style="list-style-type: none">• Deitel, P. J. & Deitel, H. M. (2016). C++How to Program. New Jersey. Prentice Hall.• Lippman, S. B. (2005). C++ Primer. Pearson Education India.• Martin, R. C. (2017). Clean architecture: a craftsman's guide to software structure and design. Prentice Hall Press.	
2Bibliografía complementaria	
<ul style="list-style-type: none">• Balagurusamy, E. (2001). Object Oriented Programming with C++. 6 edition. Tata McGraw-Hill Education.	