



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Facultad de Estudios Superiores Aragón
Plan de Estudios



Ingeniería en Computación
Computadoras y Programación

Clave	Semestre	Créditos	Área	
	1	9.0	Programación e Ingeniería de Software	
Modalidad	Curso		Tipo	Teórico
Carácter	Obligatorio			
Horas				
Semana			Semestre	
Teóricas	4.5		Teóricas	72.0
Prácticas	0.0		Prácticas	0.0
Total	4.5		Total	72.0

Seriación indicativa

Asignatura antecedente	Ninguna
Asignatura subsecuente	Programación Orientada a Objetos, Sistemas Operativos

Objetivo general: Conocer los conceptos básicos sobre la computadora y su funcionamiento, así como aprender a programar de manera estructurada sobre el lenguaje C para resolver problemas reales.

Índice temático

No.	Tema	Horas Semestre	
		Teóricas	Prácticas
1	CONCEPTOS BÁSICOS DE COMPUTACIÓN	12.0	0.0
2	METODOLOGÍA DE LA PROGRAMACIÓN ESTRUCTURADA	30.0	0.0
3	LENGUAJE C	30.0	0.0
Total		72.0	0.0
Suma total de horas		72.0	



CONSEJO ACADÉMICO DEL ÁREA DE LAS
 CIENCIAS FÍSICO MATEMÁTICAS
 Y DE LAS INGENIERÍAS

Contenido Temático	
1. CONCEPTOS BÁSICOS DE COMPUTACIÓN	
Objetivo: Conocer los conceptos básicos de las computadoras y su funcionamiento.	
1.1	Concepto de computación y computadora.
1.2	Componentes de una computadora, tableta y smartphone.
1.2.1	Hardware o componentes físicos: tarjeta madre, unidad central de proceso, memorias, sensores y periféricos.
1.2.2	Software o componentes lógicos: sistemas operativos y aplicaciones.
1.3	Computadoras analógicas y digitales.
1.4	Generaciones de computadoras digitales.
1.5	Lenguajes de programación
1.5.1	¿Qué son y para qué sirven?
1.5.2	Evolución.
1.5.3	Lenguajes declarativos.
1.5.4	Lenguajes funcionales.
1.6	Transformación de números entre sistemas decimal, binario, octal y hexadecimal.
1.7	Tipos de Procesamiento.
1.7.1	Lotes.
1.7.2	Tiempo real.
1.7.3	Tiempo Compartido.
1.7.4	Paralelo.

2. METODOLOGÍA DE LA PROGRAMACIÓN ESTRUCTURADA	
Objetivo: Resolver problemas reales con la programación estructurada.	
2.1	El paradigma de la programación estructurada.
2.2	Definición del programa: modelado de la realidad.
2.2.1	Algoritmos.
2.2.2	Abstracción.
2.2.3	Identificación de la realidad.
2.2.4	Identificación de los módulos.
2.2.5	Pseudo-código.
2.2.6	Diagrama de flujo.
2.2.7	Modularidad.
2.2.8	Concepto de caja negra.
2.3	Refinamiento sucesivo de los módulos: acoplamiento y cohesión.
2.3.1	Subcódigo y diagramas estructurados.
2.3.2	El árbol y la tabla de decisión. Instrumentación de los módulos.

3. LENGUAJE C

Objetivo: Programar de forma estructurada con el lenguaje C.

- 3.1 Elementos básicos de C.
 - 3.1.1 Función principal (main).
 - 3.1.2 Identificadores enteros y reales.
 - 3.1.3 Variables, constantes y arreglos
 - 3.1.4 Precedencia de los operadores.
 - 3.1.5 Entrada y salida de datos.
 - 3.1.6 Entrada y salida con formato.
 - 3.1.7 Caracteres y cadenas.
 - 3.1.8 Comentarios.
 - 3.1.9 Depuración.
- 3.2 Estructuras de control de programa.
 - 3.2.1 Expresiones de relación y lógicas.
 - 3.2.2 IF y ELSE.
 - 3.2.3 SWITCH.
 - 3.2.4 FOR.
 - 3.2.5 WHILE.
 - 3.2.6 DO WHILE.
- 3.3 Funciones
 - 3.3.1 Estructura de una función.
 - 3.3.2 Funciones prototipo.
 - 3.3.3 Otros tipos de variables.
 - 3.3.4 Clases de almacenamiento.
 - 3.3.5 Estructura general de un programa C.
 - 3.3.6 Macros.
 - 3.3.7 El preprocesador de C.
 - 3.3.8 Programas multiarchivo.
- 3.4 Elementos, uniones y apuntadores.
 - 3.4.1 Uniones y estructuras.
 - 3.4.2 Apuntadores a datos simples y compuestos.
 - 3.4.3 Asignación dinámica de memoria.
 - 3.4.4 Estructuras ligadas.
 - 3.4.5 Apuntadores como argumentos en funciones.
 - 3.4.6 Visibilidad.
 - 3.4.7 Operaciones sobre apuntadores.
- 3.5 Archivos
 - 3.5.1 Apertura y cierre de un archivo de datos.
 - 3.5.2 Creación de un archivo de datos.
 - 3.5.3 Procesamiento de un archivo de datos.
 - 3.5.4 Archivos de datos formateados.
- 3.6 Programación de bajo nivel
 - 3.6.1 Operadores de bits
 - 3.6.2 Campos de bits.



Estrategias didácticas		Evaluación del aprendizaje		Recursos	
Exposición	(X)	Exámenes parciales	(X)	Aula interactiva	()
Trabajo en equipo	(X)	Examen final	(X)	Computadora	(X)
Lecturas	(X)	Trabajos y tareas	(X)	Plataforma tecnológica	(X)
Trabajo de investigación	(X)	Presentación de tema	(X)	Proyector o Pantalla LCD	(X)
Prácticas (taller o laboratorio)	()	Participación en clase	(X)	Internet	(X)
Prácticas de campo	()	Asistencia	()		
Aprendizaje por proyectos	(X)	Rúbricas	()		
Aprendizaje basado en problemas	(X)	Portafolios	()		
Casos de enseñanza	()	Listas de cotejo	()		
Otras (especificar)		Otras (especificar)		Otros (especificar)	

Perfil profesiográfico	
Título o grado	<ul style="list-style-type: none"> • Poseer un título a nivel licenciatura en Ingeniería en Computación, Ciencias de la Computación, Matemáticas Aplicadas a la Computación o carreras cuyo perfil sea afín al área de Programación e Ingeniería de Software.
Experiencia docente	<ul style="list-style-type: none"> • Poseer conocimientos y experiencia profesional relacionados con los contenidos de la asignación a impartir. • Tener la vocación para la docencia y una actitud permanentemente educativa a fin de formar íntegramente al alumno: <ul style="list-style-type: none"> ○ Para aplicar recursos didácticos. ○ Para motivar al alumno. ○ Para evaluar el aprendizaje del alumno, con equidad y objetividad.
Otra característica	<ul style="list-style-type: none"> • Poseer conocimientos y experiencia pedagógica referentes al proceso de enseñanza-aprendizaje. • Tener disposición para su formación y actualización, tanto en los conocimientos de su área profesional, como en las pedagógicas. • Identificarse con los objetivos educativos de la institución y hacerlos propios. • Tener disposición para ejercer su función docente con ética profesional: <ul style="list-style-type: none"> ○ Para observar una conducta ejemplar fuera y dentro del aula. ○ Para asistir con puntualidad y constancia a sus cursos. ○ Para cumplir con los programas vigentes de sus asignaturas.

Bibliografía básica	Temas para los que se recomienda
Deitel, H. M. (2009). <i>Como programar en C++</i> . México: Pearson Educación.	1
Kernighan, B. (1991). <i>El lenguaje de programación C</i> . México: Prentice Hall.	3
Méndez, A. (2013). Diseño de algoritmos y su programación en C. México: Alfaomega.	2 y 3
Peñaloza, R. E. (2004). <i>Fundamentos de programación</i> . México: UNAM ENEP Aragón.	1, 2 y 3
Schildt, H. (1991). <i>Ansi C a su alcance</i> . México: McGraw Hill.	3
Sznajdleder, P. (2017). <i>Programación estructurada a fondo: implementación de algoritmos en C</i> . Buenos Aires: Alfaomega.	2 y 3
Wirth, N. (1976). <i>Algorithms and data structures</i> . USA: Prentice Hall.	3

Bibliografía complementaria	Temas para los que se recomienda
Alvarado, I. (2017). <i>100 problemas resueltos de programación en lenguaje C para ingeniería</i> . Madrid: Ediciones Paraninfo.	2 y 3
Long, L. (1995). <i>Introducción a las Computadoras y a los Sistemas de Información</i> . México: Prentice hall.	1, 2 y 3
Pradip, D. (2014). <i>Computer fundamentals and programming in C</i> . New Delhi: Oxford University Press.	1, 2 y 3
Restrepo, J. (1996). <i>De dos a Windows Introducción a las Computadoras Personales</i> . EU: Random House reference.	1, 2 y 3
Zapata, L. (2012). <i>Desarrollo del pensamiento analítico y sistémico: guía práctica para aprender a programar por competencias</i> . Colombia: Politecnico Colombiano Jaime Isaza Cadavid.	2