



**UNIVERSIDAD AUTÓNOMA DE YUCATÁN**  
**FACULTAD DE MATEMÁTICAS**



**MISIÓN**

Formar profesionales altamente capacitados, desarrollar investigación y realizar actividades de extensión, en Matemáticas y Computación, así como en sus diversas aplicaciones.

**FUNDAMENTOS DE PROGRAMACIÓN**

Primer semestre

**LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN**  
**LICENCIATURA EN INGENIERÍA DE SOFTWARE**  
**LICENCIATURA EN INGENIERÍA EN COMPUTACIÓN**

Agosto 2006 - Enero 2007

## **LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN**

Objetivos:

Formar profesionales calificados en el área de las ciencias de la computación para desarrollar tecnología computacional, realizar actividades de investigación, y utilizar de manera óptima sus diversas aplicaciones, con apego a la ética profesional y el servicio a la sociedad.

Objetivos específicos:

- a) Desarrollar modelos teóricos y prácticos utilizando las ciencias matemáticas y computacionales para implementar aplicaciones novedosas y eficientes.
- b) Analizar, diseñar, desarrollar e implantar software de base y de aplicaciones, utilizando o creando metodologías y ambientes computacionales, con base en la estructura, operación y necesidades de información de las organizaciones y las industrias a las que pertenecen.

## **LICENCIATURA EN INGENIERÍA EN COMPUTACIÓN**

1. Diseñar soluciones integradas de hardware y software a problemas de índoles científico y tecnológico en materia de análisis e integración de sistemas complejos.
2. Analizar e identificar los requerimientos para el diseño de sistemas computacionales acordes a la tecnología pertinente.
3. Adaptar, modificar e implementar capacidades y aplicaciones a sistemas de cómputo ad-hoc.
4. Automatizar y monitorear procesos de distinta índole, integrándolos bajo estándares de calidad y donde la alta propensión a la incertidumbre sea factor crítico.

## **LICENCIATURA EN INGENIERÍA DE SOFTWARE**

Que el egresado:

1. Explique y aplique un proceso de desarrollo de software sistémico acorde a la magnitud y complejidad de proyectos de aplicación, sean o no éstos nuevos desarrollos, tomando en cuenta la evolución y el cambio de los mismos.
2. Describa y aplique las herramientas necesarias para la especificación, diseño, verificación y validación de sistemas de software.
3. Se desempeñe en al menos un área de concentración, trabaje y se comunique de forma profesional en equipos interdisciplinarios.
4. Aplique el conocimiento y las habilidades para mejorar el proceso de desarrollo de software.
5. Contribuya al avance de la Ingeniería de Software con un acervo de conocimientos tanto teóricos como prácticos.

# Fundamentos de Programación

Semestre: Primero  
Horas: 72  
Hrs/sem: 4.5  
Créditos: 9  
Clave: AP-01

## I Generales.

**Licenciatura:** Ciencias de la Computación, Ingeniería en Computación, Ingeniería de Software.

**Clases por semana:** 4      **Número de clases:** 64      **Duración de la clase:** 90 minutos  
**Horas teóricas:** 32      **Horas prácticas:** 32

**Antecedentes académicos:** Ninguno.

**Cuerpo(s) académico responsable(s):** Diseño de aplicaciones-Ingeniería de Software.  
-Ciencias Computacionales.

**Elaboradores:** Arturo Espinosa, Otilio Santos, Alberto Marín, Víctor Menéndez, Víctor Villanueva, Edgar Cambranes, Juan Garcilazo, Raúl Duarte Achach.

**Plan de estudio aplicable:** Septiembre, 2004.

## II Descripción y justificación del curso.

El alumno implementará programas escritos en el lenguaje de programación ANSI C, aplicando conceptos de Ingeniería de Software respecto a la construcción de programas de computadoras bajo la metodología de programación estructurada.

El curso estará orientado hacia el trabajo personal bajo el ambiente de programación UNIX, aunque se revisarán al final del curso otras opciones en diferentes plataformas (Windows, sistemas empotrados). El curso tendrá sobre todo una fuerte orientación práctica, lo cual será reflejado en la forma de trabajo en clase y en la forma de evaluar el curso.

Este es el primer curso formal de programación para las licenciaturas en Ciencias de la Computación, Ingeniería de Software e Ingeniería en Computación, el cual es base para todas las otras asignaturas con fuerte contenido práctico.

## III Objetivos.

1. Construir productos de software pequeños utilizando el lenguaje de programación ANSI C.
2. Resolver problemas específicos de programación.
3. Usar prácticas probadas en codificación de programas de computadoras.

## IV Resultados esperados.

Al final del curso el estudiante debe ser capaz de:

- a) Leer, escribir y entender código en lenguaje C.
- b) Crear productos de software pequeños habiéndoles asegurado su calidad.

## V Contenido.

### Tema 1 INTRODUCCIÓN. (5 sesiones).

#### Objetivo

Describir los diferentes niveles de programación que existen (código máquina, lenguaje ensamblador, lenguajes de alto nivel), así como las diferencias y modo de funcionamiento entre interpretes y compiladores, y la mecánica del proceso de compilación. Se introducirá al alumno al uso de herramientas básicas de programación.

- i. Lenguajes de programación.
- ii. Compiladores e intérpretes.
- iii. Distribución de memoria en un programa.
- iv. El proceso de compilación.
- v. Herramientas básicas de programación.

### Tema 2 ALGORITMIA (6 sesiones)

#### Objetivo

Mediante el uso de diagramas de flujo y pseudocódigo, describir el proceso y los pasos necesarios que se deben seguir para dar solución a un problema específico.

- i. Algoritmos y programas.
- ii. Diagramas de flujo.
- iii. Seudocódigo.

### Tema 3 TIPO DE DATOS. (8 sesiones).

#### Objetivo

Aplicar los diferentes tipos de datos que se utilizan en el lenguaje de programación ANSI C, sus rangos de operación, declaración de instancias, y las operaciones aritméticas, lógicas y relacionales que se les pueden aplicar.

- i. Representación de diferentes bases, conversión entre las mismas.(DECIMAL, BINARIO OCTAL Y HEXADECIMAL).
- ii. Caracteres. (código ASCII, Unicode).
- iii. Enteros. (con o sin signo).
- iv. Punto flotante.
- v. Identificadores, constantes y variables.
- vi. Enumeraciones.
- vii. Conversión de tipos de datos (moldes).
- viii. Operadores (Aritméticos, Lógicos y Relacionales de expresiones y bits).

#### Tema 4 FUNCIONES DE ENTRADA/SALIDA. (4 sesiones).

##### Objetivo

Aplicar las funciones básicas de entrada y salida del programa, de la biblioteca estándar de ANSI C.

- i. Concepto de flujos streams. Entradas y salidas básicas de un programa: stdin, stdout y stderr.
- ii. Funciones putchar, getchar, puts y gets.
- iii. Funciones de alto nivel: printf, y scanf.
- iv. Formateado de entrada y salida usando printf, y scanf.
- v. Funciones sscanf y sprintf.

#### Tema 5 PROPOSICIONES DE CONTROL DE FLUJO DE PROGRAMA. (6 sesiones).

##### Objetivo

Aplicar las sentencias del lenguaje de programación ANSI C, que permitan la bifurcación e iteración de las instrucciones en el programa.

- i. Falso y verdadero en c.
- ii. La sentencia if / else.
- iii. La sentencia switch /case /break.
- iv. La sentencia for.
- v. La sentencia while y do ... while.
- vi. El uso de break y continue.

#### Tema 6 IMPLEMENTACIÓN DE FUNCIONES. (9 sesiones).

##### Objetivo

Aplicar los conceptos para dividir un programa en módulos, la forma de definir e invocar a funciones, así como las formas de paso de parámetros.

- i. ¿Qué es la modularidad?
- ii. Criterios para la modularidad.
- iii. Definición de funciones.
- iv. Los prototipos: definición y cuando usarlos.
- v. Reglas para la modularidad.
- vi. Principios para la construcción de software utilizando modularidad.
- vii. Invocación de funciones.
- viii. Paso de parámetros a funciones por valor.

#### Tema 7 ARREGLOS. (6 sesiones).

##### Objetivo

Aplicar los conceptos de arreglos n-dimensionales, su forma de inicialización y la relación que tienen en su estructura interna por medio de apuntadores usando ANSI C.

- i. Concepto de arreglo.
- ii. Inicialización de arreglos.
- iii. Uso de arreglos.

- iv. Arreglos multidimensionales.

#### Tema 8 CADENAS. (5 sesiones).

##### Objetivo

Aplicar el concepto de cadena, la forma de declaración que se utiliza y funciones básicas para manipulación de cadenas.

- i. Definición de cadena.
- ii. Declaración de cadenas.
- iii. Funciones básicas para manipulación de cadenas.

#### Tema 9 APUNTADORES. (12 sesiones).

##### Objetivo

Aplicar el concepto de apuntador dentro del lenguaje de programación ANSI C.

- i. Introducción al concepto de apuntador.
- ii. Declaración de apuntadores.
- iii. Desreferencias.
- iv. Aritmética de apuntadores.
- v. Apuntadores y arreglos.
- vi. Paso de parámetros a funciones por referencia.

#### Tema 10 ESTRUCTURAS Y UNIONES. (6 sesiones).

##### Objetivo

Aplicar el concepto de estructura y uniones, su declaración y uso por medio de referencias estáticas y por medio de apuntadores.

- i. Concepto de Estructura.
- ii. Declaración y uso de estructura.
- iii. Referencia a elementos de una estructura por medio de referencias estáticas y apuntadores.
- iv. Concepto de unión.
- v. Uso de uniones en programas.

## **VI Descripción del proyecto en el semestre Septiembre2005-Enero2006**

Se disponen de 6 clases para asesorías en la elaboración del proyecto final.

## **VII Estrategias de aprendizaje y enseñanza.**

La enseñanza se llevará a cabo bajo la modalidad de conferencia dictada en el aula, y por medio de sesiones prácticas en salas de cómputo. Se fomentará la participación del alumno a través de ejercicios en clase para la resolución de problemas.

## VIII Criterios de evaluación.

Método	Porcentaje	Descripción
Exámenes parciales	50%	
Tareas	20%	
Proyecto final	30%	
Exentos: A criterio del maestro		

## IX Bibliografía.

### Libros y artículos.

1. Schildt, Herbert. **Lenguaje C, Programación Avanzada**. McGraw Hill.
2. Bronson, Gary. **Algorithm Development and Program Design Using C**, PWS Publishing Co.; Book and Disk edition (February 15, 1996) ISBN: 0314069879.
3. Standish, Thomas. **Data Structures, Algorithms, and Software Principles in C**, Addison-Wesley Pub Co; 1st edition (September 30, 1994), ISBN: 0201591189.
4. Linden, Peter. **Expert C Programming**, Prentice Hall, 1994, ISBN 0131774298.
5. Reek, Kenneth. **Pointers on C**. Addison Wesley, 1997. ISBN 067399866.
6. Kernighan, Brian W. & Dennis, M. Richie. **El lenguaje de programación C**. 2ª Ed. Prentice Hall, 1988.

## X Recursos computacionales necesarios.

### Indispensable:

Pizarrón y computadoras para efectuar prácticas. La entrega de trabajos será por vía electrónica.

### Recomendable:

## XI Perfil académico del docente.

Ingeniero de Software, Licenciado en Ciencias de la Computación o carrera afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

**Modificado por:** M.I.E. Otilio Santos Aguilar.  
M. en C. Luis Basto Díaz.  
M.T.I. Edwin León Bojórquez.

**Fecha de Modificación:** 23 de Julio de 2006.