



# UNIVERSIDAD DEL VALLE DE MÉXICO

## PROGRAMA DE ESTUDIO DE LICENCIATURA

### PRAXIS MES XXI

NOMBRE DE LA ASIGNATURA: PRINCIPIOS DE PROGRAMACIÓN

FECHA DE ELABORACIÓN: MARZO 2005

ÁREA DEL PLAN DE ESTUDIOS: AS ( ) AC ( ) APOBL ( X ) APOPT ( )  
ASIGNATURA INTEGRADORA ( )

CLAVE: 532709

ASIGNATURA ANTECEDENTE: NINGUNA  
CLAVE NOMBRE

HORAS DE APRENDIZAJE A LA SEMANA		
CON DOCENTE	INDEPENDIENTES	TOTAL
3	3	6

CRÉDITOS: 5.6

TOTAL DE HORAS – CLASE POR ASIGNATURA: 90

#### OBJETIVO GENERAL

El estudiante aplicará los conocimientos básicos de programación en la solución de problemas que involucren procesamiento de datos mediante un programa de cómputo, desarrollado en algún lenguaje de programación en particular.

#### ÍNDICE DE UNIDADES

1. Introducción a la programación.
2. Estructuras de control.
3. Funciones.
4. Estructuras de datos estáticas y dinámicas.

**NÚMERO Y NOMBRE DE LA UNIDAD:**

1. Introducción a la programación

**HORAS: (10/10) 20****OBJETIVO DE LA UNIDAD:**

El estudiante identificará los elementos básicos de todo lenguaje de programación, con la finalidad de comparar las posibles diferencias que existan entre ellos en cuanto a estructura, sintaxis y procedimientos.

TEMAS Y SUBTEMAS	ESTRATEGIAS DE INSTRUCCIÓN *	EXPERIENCIAS DE APRENDIZAJE	
		Con Docente	Independientes**
1.1 Definición de algoritmos 1.2 Técnicas para elaboración de algoritmos 1.3 Representación de algoritmos 1.3.1 Diagramas de flujo 1.3.2 Pseudocódigo 1.4 Variables, Constantes y Operadores 1.4.1 Nombres de Variables 1.4.2 Tipos de Datos 1.4.3 Declaración de Variables 1.4.4 Instrucción de asignación 1.4.5 Constantes 1.4.6 Operadores 1.5 Tipos de expresiones e instrucciones 1.6 Técnicas de programación	<ul style="list-style-type: none"> <li>Exposición de la definición, técnicas y representación de algoritmos en la resolución de problemas (estrategia de recepción)</li> <li>Explicación de un programa clásico en el que se identifiquen los componentes de un programa como son; Variables, Constantes, Operadores y Expresiones de dos lenguajes de programación de alto nivel (estrategia interpersonal)</li> <li>Resolver en equipos un programa empleando el método Top-Down y exponer sus experiencias. (estrategia de proceso de grupo)</li> <li>Utilización de la computadora como apoyo didáctico (estrategia de recepción e interpersonal)</li> </ul>	<ul style="list-style-type: none"> <li>Ejercicio de análisis sobre las actividades realizadas en la solución de los problemas presentados. (DD)</li> <li>Definir las actividades necesarias en la estructuración de un algoritmo (DD, CE)</li> <li>Identificar el uso variables, constantes, operadores y expresiones para la resolución de problemas empleando el método top-down y utilizando la computadora como herramienta. (DD, MI, VG)</li> </ul>	<ul style="list-style-type: none"> <li>Analizar la solución de programas a problemas clásicos. (DD, CE)</li> <li>Resolver ejercicios en los que aplique los temas vistos y reportar resultados al docente. (DD, CE, DI, DC, VG)</li> </ul>

ESCENARIOS	ESTRATEGIAS DE EVALUACIÓN	RECURSOS DIDÁCTICOS Y/ O SOFTWARE
<ul style="list-style-type: none"> <li>Centro</li> </ul>	<ul style="list-style-type: none"> <li>Tareas (DD, CE)</li> <li>Prácticas de laboratorio (DD)</li> <li>Examen escrito (DD)</li> </ul>	<ul style="list-style-type: none"> <li>Visual C ++</li> </ul>

<b>NÚMERO Y NOMBRE DE LA UNIDAD:</b> 2. Estructuras de control		<b>HORAS:(13/12) 25</b>	
<b>OBJETIVO DE LA UNIDAD:</b> El estudiante diseñará programas de cómputo básicos utilizando estructuras de control de flujo de datos en la solución de problemas sencillos que requieren procesamiento de información.			
<b>TEMAS Y SUBTEMAS</b>	<b>ESTRATEGIAS DE INSTRUCCIÓN *</b>	<b>EXPERIENCIAS DE APRENDIZAJE</b>	
		<b>Con Docente</b>	<b>Independientes**</b>
2.1 Estructuras de Decisión 2.1.1 La construcción IF – THEN – ELSE 2.1.2 IFS Anidados 2.1.3 SWITCH / Case 2.2 Estructuras de Repetición 2.2.1 For 2.2.2 While 2.2.3 DO - WHILE	<ul style="list-style-type: none"> <li>Exposición de las estructuras de control y su aplicación óptima en la resolución de problemas (estrategia de recepción)</li> <li>Explicación de programas clásicos en los que se identifique el empleo de las estructuras de control (estrategia interpersonal)</li> <li>Resolver en equipos programas empleando las estructuras de control y exponer sus soluciones. (estrategia de proceso de grupo)</li> <li>Utilización de la computadora como apoyo didáctico (estrategia de recepción e interpersonal)</li> </ul>	<ul style="list-style-type: none"> <li>Ejercicio de análisis sobre las actividades realizadas en la solución de los problemas presentados. (DD)</li> <li>Identificar las actividades necesarias en el empleo óptimo de las estructuras de control. (DD, CE)</li> </ul>	<ul style="list-style-type: none"> <li>Analizar la solución de programas a problemas clásicos. (DD, CE)</li> <li>Resolver ejercicios en los que aplique de manera correcta las estructuras de control y reportar resultados al docente. (DD, CE, DI, DC, VG)</li> </ul>
<b>ESCENARIOS</b>	<b>ESTRATEGIAS DE EVALUACIÓN</b>	<b>RECURSOS DIDÁCTICOS Y/ O SOFTWARE</b>	
<ul style="list-style-type: none"> <li>Centro</li> </ul>	<ul style="list-style-type: none"> <li>Tareas (DD, CE)</li> <li>Prácticas de laboratorio (DD)</li> <li>Resolución de casos (DD, VG)</li> <li>Proyecto (DD, DC, VG)</li> <li>Examen escrito (DD)</li> </ul>	<ul style="list-style-type: none"> <li>Visual C++</li> </ul>	

NÚMERO Y NOMBRE DE LA UNIDAD: 3. Funciones		HORAS: (10/10) 20	
<b>OBJETIVO DE LA UNIDAD:</b> El estudiante diseñará programas de cómputo de complejidad intermedia utilizando estructuras de control, insertadas en subprogramas reutilizables que comparten información entre sí para optimizar su funcionamiento, y los elementos de la programación estructurada.			
TEMAS Y SUBTEMAS	ESTRATEGIAS DE INSTRUCCIÓN *	EXPERIENCIAS DE APRENDIZAJE	
		Con Docente	Independientes**
3.1 Funciones 3.2 Procedimientos 3.3 Correspondencia entre argumento y parámetro	<ul style="list-style-type: none"> <li>Exposición de las bondades y sintaxis del empleo de funciones y procedimientos en la resolución de problemas (estrategia de recepción)</li> <li>Explicación de programas clásicos en los que se identifique el empleo de funciones y procedimientos (estrategia interpersonal)</li> <li>Resolver en equipos programas empleando funciones, procedimientos y exponer sus soluciones. (estrategia de proceso de grupo)</li> <li>Utilización de la computadora como apoyo didáctico (estrategia de recepción e interpersonal)</li> </ul>	<ul style="list-style-type: none"> <li>Ejercicio de análisis sobre el empleo de funciones y procedimientos como el uso de argumento y parámetro en la solución de los problemas presentados. (DD)</li> <li>Identificar las actividades necesarias en la utilización de una función o procedimiento en la solución de los problemas presentados. (DD, CE)</li> </ul>	<ul style="list-style-type: none"> <li>Analizar el uso de funciones y procedimientos en programas clásicos. (DD, CE)</li> <li>Resolver ejercicios en los que aplique de manera optima el uso de funciones, procedimientos y reportar resultados al docente. (DD, CE, DI, DC, VG)</li> </ul>
ESCENARIOS	ESTRATEGIAS DE EVALUACIÓN	RECURSOS DIDÁCTICOS Y/ O SOFTWARE	
<ul style="list-style-type: none"> <li>Centro</li> </ul>	<ul style="list-style-type: none"> <li>Tareas (DD, CE)</li> <li>Prácticas de laboratorio (DD)</li> <li>Resolución de casos (DD, VG)</li> <li>Proyecto (DD, DC, VG)</li> <li>Examen escrito (DD)</li> </ul>	<ul style="list-style-type: none"> <li>Visual C++</li> </ul>	

<b>NÚMERO Y NOMBRE DE LA UNIDAD:</b> 4. Estructuras de datos estáticas y dinámicas		<b>HORAS: (12/13) 25</b>	
<b>OBJETIVO DE LA UNIDAD:</b> El estudiante diseñará programas de cómputo vinculados a estructuras de datos conocidas residentes en la memoria principal de la computadora y, dependiendo de la construcción de dicha estructura, aplicará los procedimientos requeridos para la interacción con la información contenida en ellas			
<b>TEMAS Y SUBTEMAS</b>	<b>ESTRATEGIAS DE INSTRUCCIÓN *</b>	<b>EXPERIENCIAS DE APRENDIZAJE</b>	
		<b>Con Docente</b>	<b>Independientes**</b>
4.1 Arreglos 4.1.1 Definiciones de arreglos 4.1.2 Notación del arreglo 4.1.3 Arreglos multidimensionales 4.2 Estructuras 4.2.1 Declaración de estructura 4.2.2 Variables de tipo estructura 4.2.3 Asignación de valores a variables de estructuras 4.2.4 Variables de estructuras y arreglos 4.3 Apuntadores 4.3.1 ¿Para que sirven? 4.3.2 Declaración 4.3.3 Listas secuenciales	<ul style="list-style-type: none"> <li>Exposición de arreglos, apuntadores y estructuras de datos en la resolución de problemas (estrategia de recepción)</li> <li>Explicación de programas clásicos (ordenación, búsquedas) en los que se identifique el uso de arreglos y apuntadores (estrategia interpersonal)</li> <li>Resolver en equipo el caso del manejo de una Biblioteca, una versión con arreglos y otra con apuntadores, y exponer sus soluciones. (estrategia de proceso de grupo)</li> <li>Utilización de la computadora como apoyo didáctico (estrategia de recepción e interpersonal)</li> </ul>	<ul style="list-style-type: none"> <li>Ejercicio de análisis sobre las actividades realizadas en la solución de los problemas presentados. (DD)</li> <li>Identificar las actividades necesarias en el empleo correcto de los arreglos, apuntadores y estructuras de datos. (DD, CE)</li> </ul>	<ul style="list-style-type: none"> <li>Analizar la solución de: ordenamientos, búsquedas, altas ordenadas, ABC (Altas, Bajas Consultas de cualquier programa. (DD, CE)</li> <li>Resolver ejercicios en los que aplique de manera correcta los arreglos, apuntadores y reportar resultados al docente. (DD, CE, DI, DC, VG)</li> </ul>
<b>ESCENARIOS</b>	<b>ESTRATEGIAS DE EVALUACIÓN</b>	<b>RECURSOS DIDÁCTICOS Y/ O SOFTWARE</b>	
<ul style="list-style-type: none"> <li>Centro</li> </ul>	<ul style="list-style-type: none"> <li>Tareas (DD, CE)</li> <li>Prácticas de laboratorio (DD)</li> <li>Proyecto (DD, DC, VG)</li> <li>Examen escrito (DD)</li> </ul>	<ul style="list-style-type: none"> <li>Visual C++</li> </ul>	

## REFERENCIAS BIBLIOGRÁFICAS

### BÁSICA:

Deitel, Harvey M. (2003). *Cómo programar en C/C++*. México: Pearson Education, 4a, ISBN: 970-26-0254-8

Parsons, June Jamrich (2004). *Conceptos de Computación*. México: International Thomson, 6a, ISBN: 970-686-281-1

Microsoft. (2000). *Office 2000: Introducción*, México: International Thomson. ISBN: 970-686-000-2

### COMPLEMENTARIA:

Baleana, Francesco (2003). *“Visual Basic .net”*. McGraw-Hill. ISBN 0-7356-1375-3

Hancock, Les & Krieger, Morris. (1998). *“Introducción al lenguaje C”*. McGraw-Hill. ISBN 0-07-025995-X.

Seit, Ravi (1996). *“Programming Languages, Concepts and Constructs”*, Addison-Wesley.

Schildt, Herbert (1987). *“C The Complete Referente”*. McGraw-Hill. ISBN 0-07-881263-1.

Schneider, G.M. & S.C. Bruell (1981). *“Advanced Programming and Solving with Pascal”*. Wiley

Tremblay, Jean Paul. & Bunt, Richard B. (1982). *“Introducción a la Ciencia de las Computadoras Enfoque Algorítmico”*. McGraw-Hill. ISBN 968-451-360-7.



**UNIVERSIDAD DEL VALLE DE MÉXICO**  
**PROGRAMA DE ESTUDIO DE LICENCIATURA**  
**PRAXIS MES XXI**

**ASIGNATURA:** PRINCIPIOS DE PROGRAMACIÓN

**CLAVE:** 532709

PERFIL DOCENTE							
NIVEL DE ESCOLARIDAD	PROFESIÓN	EXPERIENCIA PROFESIONAL			EXPERIENCIA DOCENTE		
		ÁREA	ACTIVIDADES	AÑOS	NIVEL EDUCATIVO	ASIGNATURAS	AÑOS Y/O SEMESTRES
Maestría Certificación Tecnológica	Lic. en Sistemas Ing. en Sistemas Ing. en Computación	Programación	Diseño de Sistemas  Implantación de Sistemas  Mantenimiento de Sistemas	2	Licenciatura	Introducción a la Programación. Programación Estructurada	4 Semestres

**OTROS CONOCIMIENTOS DESEABLES:**

Contar con al menos 6 cursos de una Maestría y / o Doctorado en el área de la asignatura.