



**PROGRAMA DE ESTUDIO POR COMPETENCIAS
PROGRAMACIÓN ESTRUCTURADA**

I. IDENTIFICACIÓN DEL CURSO

ESPACIO EDUCATIVO: Facultad de Ingeniería						
LICENCIATURA: Ingeniería en Computación				ÁREA DE DOCENCIA: Programación e Ingeniería de Software		
AÑO DE APROBACIÓN POR EL CONSEJO UNIVERSITARIO:						
APROBACIÓN POR LOS H.H. CONSEJOS ACADÉMICO Y DE GOBIERNO		FECHA:		PROGRAMA ACTUALIZADO POR: M. en I. Mireya Salgado Gallegos M. en A. Edith Salazar Vázquez M. en A. Silvia Edith Albarrán Trujillo M. en C. Eduardo Trujillo Flores Ing. Álvaro Arzate Trejo Ing. Leticia Hernández Linares		PROGRAMA REVISADO POR: Integrantes de la Academia de Programación e Ingeniería de Software
				FECHA DE ELABORACIÓN : Mayo 2007		FECHA DE REVISIÓN : Noviembre 2013
CLAVE	HORAS DE TEORÍA	HORAS DE PRÁCTICA	TOTAL DE HORAS	CRÉDITOS	TIPO DE CURSO	NÚCLEO DE FORMACIÓN
L41012	3	3	6	9	Curso-Laboratorio	Sustantivo
UNIDAD DE APRENDIZAJE ANTECEDENTE Ninguna				UNIDAD DE APRENDIZAJE CONSECUENTE Programación avanzada		
PROGRAMAS EDUCATIVOS O ESPACIOS ACADÉMICOS EN LOS QUE SE IMPARTE: Licenciatura en Ingeniería en Computación (Facultad. de Ingeniería, Centros Universitarios: Atlacomulco, Ecatepec, Texcoco, Valle de Chalco,						



Valle de México, Valle de Teotihuacán, Zumpango)

II. PRESENTACIÓN DEL PROGRAMA

El Ingeniero en Computación es el profesional que posee los conocimientos y habilidades en el desarrollo de sistemas computacionales, diseño y mantenimiento de hardware, comunicaciones y redes de computadoras así como en la administración de recursos computacionales.

Una de las principales características del Ingeniero en Computación es la habilidad que tiene para desarrollar aplicaciones que apoyan a la solución de problemas. Esta habilidad se adquiere durante la formación del profesionista e inicia desde los primeros semestres y va madurando continuamente. La programación evoluciona constantemente, sin embargo, la programación estructurada en otros paradigmas de programación es el conocimiento base y permite la adquisición de habilidades de programación que se requieren en la construcción de aplicaciones de mediana y alta complejidad.

Esta unidad de aprendizaje tiene la finalidad de introducir al alumno al ámbito de la programación en sus elementos básicos y proporcionarle los conocimientos necesarios y suficientes para que utilice estructuras de datos básicas en la programación, registros y programación modular, para la solución de problemas informáticos

III. LINEAMIENTOS DE LA UNIDAD DE APRENDIZAJE

DEL DOCENTE	DEL DISCENTE
<ul style="list-style-type: none"> • Asistencia de un 90% a clases. • Cubrir el 100% del programa. • Aplicar y seguir el programa en un 100% • Registrar la asistencia en cada clase. • Realizar el encuadre del curso. • Asesorar y conducir el trabajo de la unidad de aprendizaje. • Fomentar el intercambio de experiencias. • Retroalimentar el diseño del programa. • Resolver las dudas de los alumnos. • Evaluar la unidad de aprendizaje. 	<ul style="list-style-type: none"> • Asistir al 80% de las sesiones para tener derecho al examen ordinario. • Asistir al 60% de las sesiones para tener derecho al examen extraordinario. • Asistir al 30% de las sesiones para tener derecho al examen a título de suficiencia. • Tener sentido de responsabilidad en los trabajos extraclase. • Entregar en tiempo y forma los trabajos extraclase. • Tener sentido de integración y participación dentro del salón de clase.



<ul style="list-style-type: none">• Dar a conocer oportunamente resultados de evaluación de trabajos y exámenes• Programar y llevar a cabo la revisión de la evaluación de trabajos y exámenes• Establecer la tolerancia para el inicio de clases.• Proponer y respetar la forma de evaluación de la unidad de aprendizaje.• Generar en sus alumnos una visión integradora de la unidad de aprendizaje.• Respetar el número de horas teóricas y prácticas de la unidad de aprendizaje	<ul style="list-style-type: none">• 100% de cumplimiento de trabajos entregados antes del examen final.• Realizar y analizar la lectura de los textos propuestos.• Practicar la apertura hacia el aprendizaje.• Tener derecho a revisión de evaluaciones de trabajos y exámenes, dentro de los 5 días hábiles siguientes a la publicación de la calificación, solicitándolo por escrito.• Tener sentido de integración y participación dentro del salón de clases.• Asistir a las sesiones y ser puntual.• Realizar las actividades encomendadas y las tareas que se dejen para casa.• En caso de no asistir, realizar las actividades llevadas a cabo durante la sesión para ser entregadas al docente.
--	---

IV. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

Aplicar el paradigma de la programación estructurada para representar en términos de pseudocódigo, la solución de problemas reales automatizables, mostrando en ella el dominio de variables simples, vectores, matrices, registros y modularidad.

V. COMPETENCIAS GENÉRICAS

Desarrollo de sistemas

VI. ÁMBITOS DE DESEMPEÑO PROFESIONAL

Áreas informáticas y áreas que requieran el apoyo informático en empresas públicas y privadas.



VII. ESCENARIOS DE APRENDIZAJE

Salón de clases

VIII. ESTRUCTURA DE LA UNIDAD DE APRENDIZAJE

1. Identificar las fases de la metodología de programación estructurada para la solución de problemas.
2. Aplicar la programación estructurada en la solución de problemas utilizando diagramas de flujo y pseudocódigo.
3. Utilizar arreglos unidimensionales y bidimensionales para el almacenamiento de datos en la solución de problemas.
4. Usar las técnicas de programación modular en el desarrollo de programas informáticos.
5. Utilizar los registros para almacenar y manipular información en el desarrollo de programas informáticos.

IX. DESARROLLO DE LA UNIDAD DE APRENDIZAJE

UNIDAD DE COMPETENCIA I:	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES / VALORES
Identificar las fases de la metodología de programación estructurada para la solución de problemas	Definir y explicar: Algoritmo, fases de la solución de un programa, sentencia, identificadores, expresiones, reglas de precedencia de signos operacionales, estructuras de datos y estructuras de control.	<p>Conceptuar y ejemplificar los elementos de la programación estructurada.</p> <p>Analizar las fases de la solución de problemas de programación estructurada.</p> <p>Diseñar creativamente las diferentes soluciones de un programa informático</p>	<p>Receptiva</p> <p>Analítica</p> <p>Propositiva</p> <p>Tolerancia</p> <p>Perseverancia</p>



ESTRATEGIAS DIDÁCTICAS: Estrategias de enseñanza: Uso de mapas mentales, diagramas de flujo, mapas conceptuales, resumen, cuestionarios, casos prácticos, presentaciones en Power Point, trabajos en equipo y grupal. Estrategias de aprendizaje: Elaboración de algoritmos Lectura de textos obtenidos de investigación documental		RECURSOS REQUERIDOS: Pizarrón Videoprojector Computadora	TIEMPO DESTINADO: 4 horas
CRITERIOS DE DESEMPEÑO		EVIDENCIAS	
		DESEMPEÑO	PRODUCTOS
Diseño correcto de algoritmos Fases de la aplicación informática		Explicación e investigación del concepto de algoritmo, análisis de las características de los algoritmos y realización de algoritmos. Explicación e investigación de las fases de una aplicación informática.	Algoritmos diseñados por los alumnos. Mapa mental

UNIDAD DE COMPETENCIA II	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES VALORES
Aplicar la programación estructurada en la solución de problemas utilizando diagramas de flujo y pseudocódigo.	Caracterizar las diferentes estructuras de datos Caracterizar las diferentes estructuras de control.	Elaborar programas que utilizan estructuras de datos Solucionar problemas	Receptiva Analítica Propositiva Tolerancia



	<p>Conocer las instrucciones básicas</p> <p>Describir las características de un diagrama de flujo</p> <p>Describir las características de un pseudocódigo</p> <p>Describir los pasos para realizar una prueba de escritorio</p>	<p>utilizando estructuras de control</p> <p>Utilizar instrucciones básicas en la elaboración de programas.</p> <p>Diseñar programas utilizando diagramas de flujo.</p> <p>Diseñar programas utilizando pseudocódigo.</p> <p>Realizar la prueba de escritorio de un programa</p> <p>Diseñar creativamente las diferentes soluciones de un programa informático</p>	<p>Perseverancia</p>
<p>ESTRATEGIAS DIDÁCTICAS:</p> <p>Estrategias de enseñanza: Uso de mapas mentales, diagramas de flujo, mapas conceptuales, resumen, cuestionarios, casos prácticos, presentaciones en Power Point, trabajos en equipo y grupal.</p> <p>Estrategias de aprendizaje: Elaboración de diagramas de flujo y pruebas de escritorio. Elaboración de pseudocódigos de baja y media complejidad Elaboración de cuadro sinóptico</p>		<p>RECURSOS REQUERIDOS Pizarrón Videoprojector Computadora</p>	<p>TIEMPO DESTINADO 32 hrs.</p>
<p>CRITERIOS DE DESEMPEÑO</p>		<p>EVIDENCIAS</p>	
		<p>DESEMPEÑO</p>	<p>PRODUCTOS</p>



<p>Diseño de programas utilizando pseudocódigo Programas (70%) Uso de estructuras de control (50%) Diseño de solución (20%)</p> <p>Prueba (30%) Listado de variables (5%) Seguimiento de instrucciones (25%)</p> <p>Diseño de programas utilizando diagramas de flujo Programas (70%) Uso de estructuras de control (50%) Diseño de solución (20%)</p> <p>Prueba (30%) Listado de variables (5%) Seguimiento de instrucciones (25%)</p> <p>Estructuras de control, diagramas de flujo e instrucciones básicas</p>	<p>Solucionar problemas mediante el uso de pseudocódigo y prueba de escritorio aplicando las estructuras de control y las instrucciones básicas de programación</p> <p>Solucionar problemas mediante el uso de diagramas de flujo y prueba de escritorio aplicando las estructuras de control y las instrucciones básicas de programación</p> <p>Reafirmar las estructuras de control, simbología usada en diagramas de flujo e instrucciones básicas de programación</p>	<p>Programas con pseudocódigo y prueba de escritorio</p> <p>Programas con diagramas de flujo y prueba de escritorio</p> <p>Cuadro sinóptico</p>
---	---	---

UNIDAD DE COMPETENCIA III	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES VALORES
Utilizar arreglos unidimensionales y bidimensionales para el almacenamiento de datos en la solución de problemas	Describir las características de las estructuras de datos de tipo arreglo (Arreglos unidimensionales y Arreglos bidimensionales)	Diseñar programas utilizando las estructuras de datos de tipo arreglo.	Receptiva Analítica Propositiva Tolerancia



		Diseñar creativamente las diferentes soluciones de un programa informático	Perseverancia
ESTRATEGIAS DIDÁCTICAS: Estrategias de enseñanza: Uso de mapas mentales, diagramas de flujo, mapas conceptuales, resumen, cuestionarios, casos prácticos, presentaciones en power point, trabajos en equipo y grupal. Estrategias de aprendizaje: Realización de programas utilizando arreglos unidimensionales y bidimensionales.		RECURSOS REQUERIDOS Pizarrón Videoprojector Computadora Lenguaje de programación estructurada	TIEMPO DESTINADO 20 hrs.
CRITERIOS DE DESEMPEÑO		EVIDENCIAS	
		DESEMPEÑO	PRODUCTOS
Diseño de programas utilizando arreglos unidimensionales Uso de arreglos (50%) Diseño de solución (50%) Diseño de programas utilizando arreglos bidimensionales Uso de arreglos (50%) Diseño de solución (50%)		Solucionar problemas que requieran el uso de arreglos unidimensionales Solucionar problemas que requieran el uso de arreglos bidimensionales	Programas que utilicen arreglos unidimensionales Programas que utilicen arreglos bidimensionales

UNIDAD DE COMPETENCIA IV	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES VALORES
Usar las técnicas de programación modular en el desarrollo de programas informáticos	Caracterizar las variables globales y locales, los argumentos y parámetros formales. Conocer las características principales de la	Diseñar programas utilizando variables globales y locales en la realización de un programa modular.	Receptiva Analítica Prepositiva Tolerancia



	<p>programación modular</p> <p>Conocer la declaración y la llamada a un módulo.</p>	<p>Diseñar creativamente las diferentes soluciones de un programa informático</p>	<p>Perseverancia</p>
<p>ESTRATEGIAS DIDÁCTICAS:</p> <p>Estrategias de enseñanza: Uso de mapas mentales, diagramas de flujo, mapas conceptuales, resumen, cuestionarios, casos prácticos, presentaciones en power point, trabajos en equipo y grupal</p> <p>Estrategias de aprendizaje: Realización de programas con uso de menús y módulos, utilizando variables locales y globales.</p>		<p>RECURSOS REQUERIDOS Pizarrón Videoprojector Computadora Lenguaje de programación estructurada</p>	<p>TIEMPO DESTINADO 22 hrs.</p>
<p align="center">CRITERIOS DE DESEMPEÑO</p>		<p>EVIDENCIAS</p>	
		<p>DESEMPEÑO</p>	<p>PRODUCTOS</p>
<p>Diseño correcto de programas utilizando programación modular Uso de módulos (50%) Diseño de solución (50%)</p>		<p>Diseñar programas que utilicen variables globales y locales, argumento formal y actual, declaración y llamada a módulos. Utilizar problemas que puedan ser vistos como pequeños problemas independientes y funcionales por separado y la unión de todos para solucionar problemas complejos aplicando la programación modular</p>	<p>Problemas resueltos que involucren menús con varias opciones y cuya solución tenga por lo menos dos módulos con manejo de argumentos</p>



UNIDAD DE COMPETENCIA V	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES VALORES
Utilizar los registros para almacenar y manipular información en el desarrollo de programas informáticos	Describir las características de las estructuras de datos de tipo registro (registro simple, registro de registro, arreglos de registro).	Diseñar programas utilizando las estructuras de datos de tipo registro. Diseñar creativamente las diferentes soluciones de un programa informático	Receptiva Analítica Prepositiva Tolerancia Perseverancia
ESTRATEGIAS DIDÁCTICAS: Estrategias de enseñanza: Uso de mapas mentales, diagramas de flujo, mapas conceptuales, resumen, cuestionarios, casos prácticos, presentaciones en power point, trabajos en equipo y grupal. Estrategias de aprendizaje: Realización de programas con uso de registros		RECURSOS REQUERIDOS Pizarrón, proyector de cañón, computadora	TIEMPO DESTINADO 12 hrs.
CRITERIOS DE DESEMPEÑO		EVIDENCIAS	
		DESEMPEÑO	PRODUCTOS
Diseño de programas utilizando registros Uso de registros (50%) Diseño de la solución (50%)		Diseñar programas que requieran el uso de registros modelando problemas reales.	Programas en pseudocódigo
Diseño de programas utilizando arreglos de registros Uso de registros (50%) Diseño de la solución (50%)		Diseñar programas que requieran el uso de arreglos de registros modelando problemas reales.	Programas en pseudocódigo



<p>Diseño de programas utilizando registros de registros Uso de registros (50%) Diseño de la solución (50%)</p>	<p>Diseñar programas que requieran el uso de registros de registros modelando problemas reales.</p>	<p>Programas en pseudocódigo</p>
---	---	----------------------------------

X. EVALUACIÓN Y ACREDITACIÓN

<p>La evaluación del curso incluye lo siguiente:</p> <ul style="list-style-type: none"> • Evaluación diagnóstica • Evaluación formativa. • Evaluación sumaria (examen). 																			
<p>Evaluación diagnóstica: Al inicio del semestre le sirve al profesor para reafirmar algunos temas básicos para el inicio del curso.</p> <p>Evaluación formativa: Se lleva a cabo durante todo el semestre y considera la realización de productos obtenidos en las sesiones de clase y extractase. El profesor determinara la ponderación de los productos obtenidos englobados en los rubros siguientes:</p> <table border="0" data-bbox="218 889 1898 1029"> <tr> <td>Tareas</td> <td>30%</td> <td>ORDINARIO</td> <td>EXTRAORDINARIO Y TÍTULO</td> </tr> <tr> <td>Exámenes parciales</td> <td>40%</td> <td>70% semestre</td> <td>100% examen departamental</td> </tr> <tr> <td>Prácticas escritas</td> <td>30%</td> <td>30% examen departamental</td> <td></td> </tr> <tr> <td></td> <td>100%</td> <td></td> <td></td> </tr> </table> <p>Exento queda a criterio del profesor.</p>				Tareas	30%	ORDINARIO	EXTRAORDINARIO Y TÍTULO	Exámenes parciales	40%	70% semestre	100% examen departamental	Prácticas escritas	30%	30% examen departamental			100%		
Tareas	30%	ORDINARIO	EXTRAORDINARIO Y TÍTULO																
Exámenes parciales	40%	70% semestre	100% examen departamental																
Prácticas escritas	30%	30% examen departamental																	
	100%																		
<p>Acreditación. Para acreditar la unidad de aprendizaje se deberá cumplir con lo siguiente:</p> <ul style="list-style-type: none"> • Cubrir el 80% de asistencia como mínimo para calificación ordinaria, 60% para extraordinaria y 30% para título de suficiencia. • Para calificación ordinaria deberá presentar todos los exámenes parciales, y el final (salvo casos de alumnos exentos). • La calificación mínima aprobatoria será de 6.0 																			
<p>Calificación Extraordinaria o a Título de suficiencia: Examen departamental escrito sobre todo el contenido del curso y aprobado con 6.0 100%</p>																			



XI. REFERENCIAS

BIBLIOGRAFÍA BÁSICA

- ✓ Albarrán, T. S. E. & Salgado, G. M. (2008). Programación Estructurada. Universidad Autónoma del Estado de México.
- ✓ Cairó, O. (2006). Fundamentos de Programación. Piensa en C. Pearson Educación.
- ✓ Joyanes, A. L., & Zahonero, M. I. (2003). Programación en C, Metodología, Estructura de Datos y Objetos. México. McGraw-Hill.
- ✓ García, B., & Giner, J. R. (2008). Programación Estructurada en C. Madrid. Pearson Educación.

BIBLIOGRAFÍA COMPLEMENTARIA

- ✓ Balcázar, J. L. (1993). Programación metódica. España: McGraw-Hill.
- ✓ Cairó, O. (1995). Metodología de Programación, Tomo I. Colombia. Computec.
- ✓ Cairó, O. (1995). Metodología de Programación, Tomo II. Colombia. Computec.
- ✓ Carrasco, R.; Patiño, I.; Santos, M. (2010). Fundamentos de Programación (2ª ed). Ra-Ma, Alfaomega
- ✓ Criado, A. (2006). Programación en lenguajes estructurados. México. Alfaomega-Ra ma.
- ✓ Harel, D. (1987). Algorithmics, the Spirit of Computing. Massachusetts. Addison Wesley.
- ✓ Joyanes A. L. (2008). Fundamentos de programación. Algoritmos, Estructura de datos y objetos (4ª ed.). México. McGraw-Hill.
- ✓ Joyanes A. L., Castillo, S. A., Sánchez, G. L, & Zahonero, M. I. (2002). Programación en C. Libro de problemas. México. McGraw-Hill.
- ✓ Norton, P. (2006). Introducción a la Computación (6ª ed.). México. McGraw-Hill.