Universidad Autónoma del Estado de México
UAEM

PROGRAMA DE ESTUDIOS POR COMPETENCIAS

PROGRAMACIÓN AVANZADA

I. IDENTIFICACIÓN DEL CURSO

<table>
<tr><td colspan="7">ORGANISMO ACADÉMICO: Facultad de Ingeniería</td></tr>
<tr><td colspan="3">PROGRAMA EDUCATIVO: Ingeniería en Computación</td><td colspan="4">ÁREA DE DOCENCIA: Programación e Ingeniería del Software</td></tr>
<tr><td colspan="2">APROBACIÓN POR LOS H.H. CONSEJOS ACADÉMICO Y DE GOBIERNO</td><td colspan="2">FECHA:</td><td colspan="2">PROGRAMA ELABORADO POR:
Dra. Lilia Ojeda Toche
Ing. Alvaro Arzate Trejo
Ing. Élfego Gutiérrez Ocampo
PROGRAMA ACTUALIZADO POR:
Ing. Alvaro Arzate Trejo
Ing. Élfego Gutiérrez Ocampo</td><td>PROGRAMA REVISADO POR:
Integrantes de la Academia de Programación e Ingeniería de Software

Centro Universitario Valle de Chalco, Centro Universitario Zumpango</td></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2">FECHA DE ELABORACIÓN : Junio 2007
FECHA DE ACTUALIZACIÓN : Octubre 2013</td><td>FECHA DE REVISIÓN :
Octubre 2013</td></tr>
<tr><td>CLAVE</td><td>HORAS DE TEORÍA</td><td>HORAS DE PRÁCTICA</td><td>TOTAL DE HORAS</td><td>CRÉDITOS</td><td>TIPO DE UNIDAD DE APRENDIZAJE</td><td>NÚCLEO DE FORMACIÓN</td></tr>
<tr><td>L41053</td><td>3</td><td>3</td><td>6</td><td>9</td><td>Curso</td><td>Obligatoria</td></tr>
<tr><td colspan="2">PRERREQUISTOS:
Programación estructurada, Estructura de Datos</td><td colspan="2">UNIDAD DE APRENDIZAJE ANTECEDENTE:
Programación Estructurada</td><td colspan="3">UNIDAD DE APRENDIZAJE CONSECUENTE:
Ninguna</td></tr>
<tr><td colspan="7">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)</td></tr>
</table>

II. PRESENTACIÓN

Una vez adquiridas las habilidades de programación básicas bajo el paradigma estructurado, el alumno debe conocer otros paradigmas como la programación modular y la programación recursiva.

Junto con estos paradigmas el alumno se adentra en cuestiones de algorítmica, con temas de análisis y diseño de algoritmos: funcionamiento y orden de complejidad de los métodos de ordenamiento y búsqueda, técnicas de diseño como algoritmos voraces, algoritmos divide y vencerás, programación dinámica, algoritmos vuelta atrás, algoritmos ramifica y poda.

Esta formación permitirá al futuro ingeniero enfrentarse a retos de programación de alta complejidad con la certeza de poder no sólo dar una solución a un problema dado sino de dar la solución óptima y ser capaz de evaluar que tan buena es la solución dada.

La estructura planteada consta de tres unidades de competencia. La primera estudia el paradigma de programación modular y programación recursiva. La segunda revisa y analiza diferentes algoritmos de ordenamiento y búsqueda haciendo uso de la notación asintótica. La tercera unidad de competencia se dedica a varias técnicas de diseño de algoritmos.

La evaluación debe considerar tanto la parte teórica como la práctica. La primera a través de exámenes escritos y la segunda por medio del planteamiento, codificación y ejecución de algoritmos aplicando los conocimientos teóricos obtenidos en el curso.

III. LINEAMIENTOS DE LA UNIDAD DE APRENDIZAJE

DEL DOCENTE	DEL DISCENTE
• Establecer las políticas del curso al inicio del mismo. • Respetar el horario del curso y la forma de evaluarlo. • Cumplir el temario y el número de horas asignadas al curso o justificar la ausencia por adelantado (asistencia a conferencias, etc.) • Asesorar y guiar el trabajo de las unidades de aprendizaje. • Retroalimentar el trabajo de los alumnos. • Fomentar la creatividad en los alumnos a través del desarrollo de proyectos.	• Contar con la asistencia establecida en el reglamento de Facultades: o 80% para examen ordinario o 60% para examen extraordinario o 30% para examen a titulo de suficiencia • Cumplir con las actividades encomendadas entregando con calidad en tiempo y forma los trabajos requeridos. • Participar activa y críticamente en el proceso de enseñanza-aprendizaje.

• Evaluar y Calificar a los alumnos. • Preparar el material didáctico para las clases y prácticas.	• Hacer uso adecuado de las instalaciones y equipo de cómputo. • Realizar las evaluaciones que se establezcan. • Mantener unas pautas de comportamiento socialmente aceptables cuando se encuentre en clases y laboratorio. • Cuando se requiera, entregar a tiempo y forma los trabajos requeridos.

IV. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

Servir de enlace entre el aprendizaje de los paradigmas estructurado y orientado a objetos, a través de la programación modular. Presentar al alumno técnicas de programación avanzada como la recursividad. Proporcionar las habilidades necesarias para evaluar la complejidad de un algoritmo de ordenamiento o de búsqueda, así como estrategias para resolver problemas de alta complejidad, mediante técnicas de diseño avanzadas.

V. COMPETENCIAS GENÉRICAS

Tal y como se establece en el apartado 4.2.1.1 Saberes del Plan Flexible 2004 por Competencias

- Analizar y diseñar sistemas de información
- Comunicarse con expertos de otras áreas
- Utilizar eficazmente los lenguajes de programación, dispositivos electrónicos y sistemas comerciales de vanguardia
- Responder eficazmente a nuevas situaciones informáticas
- Realizar investigación de tecnología de punta
- Analizar soluciones del entorno y problemas propios de ser tratados mediante sistemas computacionales
- Aplicar los conocimientos en la práctica
- Crear nuevas ideas para la solución de problemas
- Desarrollar la habilidad análisis y síntesis de información

Algunas de estas competencias se adquieren en conjunto al estudiar el resto de unidades de aprendizaje bajo el área de competencia de Programación e Ingeniería del Software.

VI. ÁMBITOS DE DESEMPEÑO

- Empresas públicas y privadas
- Investigación de nuevas soluciones computacionales
- Docencia a cualquier nivel de aprendizaje escolarizado
- Desarrollo de proyectos.
- Análisis, diseño, implementación y mantenimiento de sistemas computacionales

VII. ESCENARIOS DE APRENDIZAJE

- Salón de Clases
- Sala de cómputo

VIII. ESTRUCTURA DE LA UNIDAD DE APRENDIZAJE

1. Paradigmas de programación modular y recursiva.
2. Análisis de algoritmos de ordenamiento y búsqueda
3. Diseño de algoritmos empleando diversas técnicas.

IX. DESARROLLO DE LA UNIDAD DE APRENDIZAJE

UNIDAD DE COMPETENCIA I	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Paradigmas de programación modular y recursiva.	- Programación modular (Análisis de las Estructuras de Control) - Programación Recursiva	Desarrollar algoritmos bajo los paradigmas de: Programación modular	Receptiva Analítica Responsabilidad para cumplir con las tareas asignadas

		Programación recursiva.	Tolerancia y participación Desarrollar la capacidad analítica ante nuevos problemas Respetar al docente y a los compañeros mediante un comportamiento socialmente aceptable
ESTRATEGIAS DIDÁCTICAS: – Presentaciones acompañadas de apuntes preparados por el profesor. – Revisión y análisis de material bibliográfico – Solución de ejercicios – Desarrollo de prácticas de programación en laboratorio		**RECURSOS REQUERIDOS** Pizarrón, Libro de texto y apuntes del docente Laboratorio de prácticas con un lenguaje de programación modular Videoproyector	**TIEMPO DESTINADO** 12 horas teóricas 12 horas práctica
CRITERIOS DE DESEMPEÑO I		**EVIDENCIAS**	
		DESEMPEÑO	**PRODUCTOS**
Resolución de problemas Práctica de laboratorio		Ejercicios teóricos Programas	Programación Modular y programación recursiva. Programación Modular y programación recursiva. Lenguaje de programación

<table>
<tr><td rowspan="2">UNIDAD DE COMPETENCIA II</td><td colspan="3">ELEMENTOS DE COMPETENCIA</td></tr>
<tr><td>CONOCIMIENTOS</td><td>HABILIDADES</td><td>ACTITUDES/ VALORES</td></tr>
<tr><td>Análisis de algoritmos de ordenamiento y búsqueda</td><td>- Orden de complejidad de un algoritmo (notación asintótica)
- Método de la burbuja
- Método de Selección
- Método de Inserción
- Método de Ordenamiento Rápido
- Método de mezclas
- Búsqueda Secuencial
- Búsqueda binaria</td><td>• Conocer el funcionamiento de un método de ordenamiento y búsqueda
• Obtener el orden de complejidad de un algoritmo de ordenamiento y búsqueda.</td><td>Receptiva
Analítica
Responsabilidad para cumplir con las tareas asignadas
Tolerancia y participación
Desarrollar la capacidad analítica ante nuevos problemas
Respetar al docente y a los compañeros mediante un comportamiento socialmente aceptable</td></tr>
<tr><td colspan="2">ESTRATEGIAS DIDÁCTICAS:
– Presentaciones acompañadas de apuntes preparados por el profesor.
– Revisión y análisis de material bibliográfico
– Solución de ejercicios
– Desarrollo de prácticas de programación en laboratorio</td><td>RECURSOS REQUERIDOS
Pizarrón,
Libro de texto y apuntes del docente
Laboratorio de prácticas con un leguaje de programación
Videoproyector</td><td>TIEMPO DESTINADO
18 horas teóricas
18 horas práctica</td></tr>
<tr><td rowspan="2">CRITERIOS DE DESEMPEÑO II</td><td colspan="2">EVIDENCIAS</td></tr>
<tr><td>DESEMPEÑO</td><td>PRODUCTOS</td></tr>
<tr><td>Resolución de problemas</td><td>Ejercicios teóricos</td><td>Funcionamiento y órdenes de complejidad de los métodos de</td></tr>
</table>

		ordenamiento y búsqueda.
Práctica de laboratorio	Programas	Funcionamiento y órdenes de complejidad de los métodos de ordenamiento y búsqueda. Lenguaje de programación.

UNIDAD DE COMPETENCIA III	**ELEMENTOS DE COMPETENCIA**		
	CONOCIMIENTOS	**HABILIDADES**	**ACTITUDES/ VALORES**
Técnicas de Diseño de algoritmos.	Estrategias de diseño de algoritmos: - Ecuaciones de Recurrencia - Algoritmos voraces - Divide y vencerás - Programación dinámica - Vuelta atrás - Ramifica y poda	Desarrollar algoritmos empleando diferentes técnicas de diseño dependiendo de la naturaleza del problema.	Receptiva Analítica Responsabilidad para cumplir con las tareas asignadas Tolerancia y participación Desarrollar la capacidad analítica ante nuevos problemas Respetar al docente y a los compañeros mediante un comportamiento socialmente aceptable
ESTRATEGIAS DIDÁCTICAS: – Presentaciones acompañadas de apuntes preparados por el profesor. – Revisión y análisis de material bibliográfico		**RECURSOS REQUERIDOS** Pizarrón, Libro de texto y apuntes del	**TIEMPO DESTINADO** 18 horas teóricas 18 horas práctica

– Solución de ejercicios – Desarrollo de prácticas de programación en laboratorio	docente Laboratorio de prácticas con un lenguaje de programación modular Videoproyector	
CRITERIOS DE DESEMPEÑO III	**EVIDENCIAS**	
	DESEMPEÑO	**PRODUCTOS**
Resolución de problemas	Desarrollo de aplicaciones utilizando las diferentes estrategias de diseño de algoritmos	Estrategias de diseño de Algoritmos

X. EVALUACIÓN Y ACREDITACIÓN

Para tener derecho a examen ordinario requiere al menos un 80% de asistencia.
Para tener derecho a presentar el examen extraordinario se requiere al menos un 60 % de asistencia.
Para tener derecho a presentar el examen a título de suficiencia se requiere al menos un 30% de asistencia.

Se realizarán dos exámenes parciales con valor del 60% (30% cada uno)
y prácticas de programación con un valor de 40%

Si la calificación es mayor o igual a 80% el alumno exenta.

La calificación final ordinaria estará compuesta por 60 % de examen escrito final y un 40 % de un proyecto acumulativo práctico

La calificación final extraordinaria estará compuesta por 70 % del examen escrito final correspondiente y un 30 % de un proyecto final práctico

La calificación final a título de suficiencia estará compuesta al 100 % del examen escrito final respectivo.

XI. REFERENCIAS

BIBLIOGRAFÍA BÁSICA:

- ✓ Base, S.; Van Gelder, A.(2002). "Algoritmos Computacionales: Introducción al análisis y diseño" Ed. Addison Wesley
- ✓ Bovet, D. P.; Crescenci, P. (2006). "*Introduction to the theory of complexity*" Ed. Creative Commons
- ✓ Brassard, G.; Bratley, P. (1997). "Fundamentos de Algoritmia", Ed. Prentice Hall.
- ✓ Cairó, Osvaldo y Guardati, Silvia. (2006). *Estructuras de datos* (3a. Edición). McGraw-Hill.
- ✓ Lee R. Teng. S. Chang R. y Tsai Y. (2007). *Introducción al diseño de algoritmos*. McGraw- Hill
- ✓ Levitin Anany (2002). *Introduction to Design and Analysis of Algoritms.* Addison Wesley.

BIBLIOGRAFÍA COMPLEMENTARIA:

- ✓ Drozdeck, Adam. (2007). *Estructuras de datos y algoritmos en Java* (2ª Edición). Thomson.
- ✓ Joyanes, Luis. M. Fernández, L: Sánchez, I. Zahonero. (2005). *Estructuras de datos en C.* McGraw-Hill. Schaum.
- ✓ Koffman, Elliot y Wolfgang, Paul. (2008). *Estructura de datos con C++*. Objetos, abstracciones y diseño. McGraw-Hill.