UNIVERSIDAD NACIONAL EXPERIMENTAL DEL TACHIRA

VICERRECTORADO ACADÉMICO

COORDINACIÓN DE INGENIERIA EN INFORMÁTICA

PROGRAMA DE DISEÑO DE ALGORITMOS

				Especialidad: Ingeniería Informática			
Código	Nombre de la Asignatura	Estrategias Metodológicas	Unidades Crédito	Densidad Horaria			Pre-requisito
				H.T.	H.P	T. hrs/sem.	
8201	Diseño de Algoritmos	Ver anexo	3	12	6	18	8501

Unidad I.: Algoritmia Elemental.				
Objetivo General: Desarrollar e introducir los conceptos básicos en el diseño de algoritmos computacionales.				
SEMANA	**OBJETIVOS ESPECÍFICOS**	**CONTENIDOS**	**ACTIVIDADES**	**EVALUACIÓN**
Semana 1	1.- Presentar y analizar el contenido programático y las estrategias de evaluación. 2.- Explicar en forma resumida los conceptos básicos para el diseño de algoritmos. 3. - Analizar en forma resumida los métodos de diagramación de algoritmos y las estructuras de control empleadas en el diseño.	1.Presentación y análisis del contenido programático y estrategias de evaluación 2.Conceptos básicos: 2.1- Variables y constantes. 2.2.- Expresiones aritméticas. 2.3.- Expresiones relaciónales. 3.Métodos de diagramación y estructuras de control: 3.1.- Seudo código, Diagrama de flujo y Diagrama estructurado. 3.2.- Enunciados de Lectura, Escritura y Asignaciones. 3.3.- Secuencias y decisiones.	Explicación didáctica y participativa con ayuda visual.	
Semana 2	1.- Analizar en forma resumida los métodos de diagramación de algoritmos y las estructuras de control empleadas en el diseño. 2.- Realizar sesiones practicas para desarrollar de la lógica de diseño Utilizando las estructuras básicas.	1.- Métodos de diagramación y estructuras de control 1.1.- Bucles o ciclos (Repita para. Repita mientras y Repita hasta). 1.2.- Contadores, Acumuladores y Centinelas. 2.- Desarrollar algoritmos con: 2.1.- Enunciados secuenciales con enunciados de asignación. 2.2.- Enunciados de decisión. 2.3.- Enunciados de repetición (bucles) simples y anidados. 2.4.- Uso de contadores, acumuladores y centinelas.	- Explicación didáctica y participativa con ayuda visual. - Ejercicios prácticos En grupo.	

Semana 3	1.- Realizar sesiones practicas para desarrollar de la lógica de diseño utilizando las estructuras básicas.	1.- Desarrollar algoritmos con: 1.1.- Enunciados secuenciales con enunciados de asignación. 1.2.- Enunciados de decisión. 1.3.- Enunciados de repetición (bucles) simples y anidados. 1.4.- Uso de contadores, acumuladores y centinelas.	- Ejercicios prácticos En grupo.	
Semana 4	1.- Realizar sesiones practicas para desarrollar de la lógica de diseño utilizando las estructuras básicas.	1.- Desarrollar algoritmos con: 1.1.- Enunciados secuenciales con enunciados de asignación. 1.2.- Enunciados de decisión. 1.3.- Enunciados de repetición (bucles) simples y anidados. 1.4.- Uso de contadores, acumuladores y centinelas.	- Ejercicios prácticos En grupo.	

Unidad II.: Técnicas de diseño de algoritmos				
Objetivo General: Analizar los métodos mas comunes para el diseño sistematizado de algoritmos.				
SEMANA	**OBJETIVOS ESPECÍFICOS**	**CONTENIDOS**	**ACTIVIDADES**	**EVALUACIÓN**
Semana 5	1.- Estudiar y analizar el uso de las estructuras de control básicas en el desarrollo de algoritmos para el manejo de vectores.	1.- Algoritmos sobre vectores: Búsqueda de un elemento en un vector: 1.1- Búsqueda secuencial. 1.2- Búsqueda binaria. 1.3- Búsqueda con centinela.	Explicación didáctica y participativa con ayuda visual. Ejercicios prácticos En grupo.	Prueba escrita de 100 Pts. Unidad 1 (1er. Parcial) Ponderación 25% En horas fuera de clase (Sábado)
Semana 6	1.- Estudiar y analizar los métodos de ordenación básicos mas comunes. 2. - Introducción al manejo de matrices utilizando las estructuras básicas.	1.- Algoritmos sobre vectores: Ordenación por selección de los elementos de un vector: 1.1- Ordenación por selección. 1.2- Ordenación por inserción directa. 1.3- Ordenación por burbuja. 2.- Algoritmos sobre matrices.	- Explicación didáctica y participativa con ayuda visual. - Ejercicios prácticos En grupo.	
Semana 7	1.- Realizar sesiones practicas para desarrollar la lógica de diseño.	1.- Analizar y diseñar algoritmos para: 1.1.- Lectura y carga de arreglos. 1.2.- Recorrido de arreglos. 1.3.- Búsqueda de elementos sobre arreglos. 1.4.- Ordenación de elementos de arreglos. 1.5.- Manejo de índices en arreglos.	- Dinámica de grupo.	

Unidad III: Técnicas de diseño básicos y Algoritmos Elaborados.				
Objetivo General: Métodos básicos de diseño y análisis de algoritmos elaborados para la solución de problemas propuestos.				
SEMANA	**OBJETIVOS ESPECÍFICOS**	**CONTENIDOS**	**ACTIVIDADES**	**EVALUACIÓN**
Semana 8	1.- Estudiar la metodología de divide y vencerás.	1.- Recursividad. 2.- Introducción al esquema divides y vencerás. 2.1- Concepto y filosofía. 2.2- Ordenación por fusión (merge). 2.3- Ordenación rápida (quicksort).	-Explicación didáctica y participativa con ayuda visual. Ejercicios prácticos.	
Semana 9	1.- Realizar sesiones practicas para ejecutar corridas mediante métodos de divide y vencerás.	1. Realizar corridas en frío sobre vectores utilizando los métodos: 1.1.-Merge. 1.2.-Quicksort	- Ejercicios prácticos grupo.	
Semana 10	1.- Introducción a los algoritmos voraces.	1. Introducción al esquema voraz y casos de estudio: 1.1- Concepto y filosofía. 1.2- Desglose de moneda. 1.3- Caminos mas cortos desde un solo origen.	- Explicación didáctica y participativa con ayuda visual. - Ejercicios prácticos.	

Unidad IV: Introducción al Lenguaje de programación C y complejidad de algoritmos.				
Objetivo General: Introducción al lenguaje C y analizar las medidas de eficiencia en el diseño de algoritmos.				
SEMANA	**OBJETIVOS ESPECÍFICOS**	**CONTENIDOS**	**ACTIVIDADES**	**EVALUACIÓN**
Semana 11	1.- Introducir los conceptos básicos sobre el lenguaje de programación C.	1.-Lenguaje C: 1.1.- Introducción al C, Compiladores vs interpretes y Estructura de un programa. 2.- Variables: 2.1.- Variables globales y locales. 2.2.- Modificadores de tipo y de acceso. 2.3.- Declaración de arreglos. 2.4.- Especificadores de clase de almacena-miento: Extem, Static, Register. 2.5.- Conversión de tipo. 2.6.- Incremento/decrcmento de variables. 2.7.- Operadores relaciónales/lógicos. 2.8.- Precedencias y Moldes.	-Explicación didáctica y participativa con ayuda visual. -Ejercicios prácticos.	Prueba escrita de 100 Pts. Unidad II (2do. Parcial) Ponderación 30% En horas fuera de clase (Sábado)
Semana 12	1.- Introducción a las instrucciones de entrada y salida. 2.- Introducción a las instrucciones de control.	1.- Instrucciones de E/S: 1.1.- Printf, Scanf, fopen, fprintf, fscanf, fclose, fread, fwrite. 1.2.- Formatos de salida: %d, %f, %s, etc. 2.- Instrucciones de Control: 2.1.- if-else-if, la alternativa ?, switch. 2.2.- Estructuras for, while, do/while, break, continué, exit, goto.	-Explicación didáctica y participativa con ayuda visual. -Ejercicios prácticos.	
Semana 13	1.- Introducción al uso de funciones, apuntadores y cadenas de carácter.	1.-Funciones: 1.1.-Argumentos. 1.2.- Llamadas por valor y referencia. 2.- Estructuras y archivos. 3.- Manejo de cadenas de carácter. 4.- Apuntadores.	- Explicación didáctica y participativa con ayuda visual. - Ejercicios prácticos	

Semana 14	1.- Verificación de la complejidad de Algoritmos. 2.- Analizar la complejidad de los algoritmos funcionales e imperativos.	1.- Complejidad algorítmica:. 1.1- Principios de inducción. 1.2- Evaluación de la eficiencia. 1.3- Notaciones para expresar la complejidad en tiempo OQ, 1.4- Propiedades de la notación OQ. 1.5- Funciones de complejidad en tiempos mas usuales. 2.- Análisis de complejidad: 2.1.- Análisis de complejidad de algoritmos no recursivos.	-Explicación didáctica y participativa con ayuda visual. -Ejercicios prácticos.	
Semana 15	1.- Analizar la complejidad de los algoritmos funcionales e imperativos.	1.- Análisis de complejidad: 1.1.- Análisis de complejidad de algoritmos recursivos.	-Explicación didáctica y participativa con ayuda visual. -Ejercicios prácticos	
Semana 16	1.- Consultas y asesorías a los alumnos.			Prueba escrita de 100 Pts. Unidad III (3er. Parcial) Ponderación 25%

LABORATORIO DISEÑO DE ALGORITMOS

Laboratorio: Manejo del entorno de programación.				
Objetivo General: Aprender el uso de las herramientas que proporciona un ambiente programable.				
SEMANA	**OBJETIVOS ESPECÍFICOS**	**CONTENIDOS**	**ACTIVIDADES**	**EVALUACIÓN**
Semana ly2	1.- Introducción al entorno de Programación. 2.- Codificar, compilar, correr y depurar algoritmos sencillos con las instrucciones básicas de E/S y de Decisión.	1.- Inducción al dos, Windows y al editor de programas. 2.- Codificar programas que hagan uso de las siguientes instrucciones: 2.1. Instrucciones de E/S: Printf, Scanf. 2.2. Instrucciones de decisión (if-else).	-Ejercicios en el Laboratorio.	
Semana 3 y 4	1.- Codificar, compilar, correr y depurar algoritmos mas complejos con las instrucciones básicas de E/S y de control.	1.- Codificar programas que hagan uso de las siguientes instrucciones: 1.1-Instrucciones de E/S: Printf, Scanf. 1.2- Instrucciones de ciclos de control mas comunes: for, while, do-while. 2. Diseñar y codificar programas en C de ejercicios propuestos que utilicen las estructuras de control.	Ejercicios en el Laboratorio.	
Semana 5 y 6	1.- Diseñar y codificar programas en Lenguaje C donde se utilicen las estructuras de control para el manejo de vectores.	1.- Diseñar, codificar y compilar programas en Lenguaje C de ejercicios pospuestos sobre: 1.1- Carga de vectores. 1.2- Búsqueda y ordenación de vectores.	Ejercicios en el Laboratorio.	-Quiz de Laboratorio 30 pts

Semana 7 y 8	1.- Diseñar y codificar programas en Lenguaje C donde se utilicen las estructuras de control para el manejo de vectores y matrices.	1.- Diseñar, codificar y compilar programas en Lenguaje C de ejercicios pospuestos sobre: 1.1- Carga de vectores y matrices. 1.2- Búsqueda y ordenación de vectores y matrices.	Ejercicios en el Laboratorio.	
Semana 9 v 10	1.- Codificar un caso de estudio de algoritmos voraces.	1.- Codificar, compilar y ejecutar el programa para el algoritmo de desglose de moneda.	Ejercicios en el Laboratorio.	
Semana 11 y 12	1.- Diseñar y codificar programas en Lenguaje C para el manejo de archivos, funciones y selección múltiple.	1.- Codificar, compilar y ejecutar programas que utilicen archivos, funciones y apuntadores.	Ejercicios en el Laboratorio.	
Semana 13 y 14	1.- Diseñar y codificar programas en Lenguaje C para manipular cadenas de carácter y apuntadores.	1.- Codificar, compilar y ejecutar programas para manejar cadenas de carácter y Apuntadores.	Ejercicios en el Laboratorio.	
Semana 15 y 16	Evaluación de laboratorio.			-Quiz de Laboratorio 40 pts

EVALUACIONES

Primer pardal: Teoría Ponderación 25%
Prueba de 100 puntos. Fecha: 11/03/2000

Segundo parcial: Teoría Ponderación 30%
Prueba de 100 puntos. Fecha: 22/04/200

Tercer parcial: Teoría Ponderación 25%
Prueba de 100 puntos. Fecha: 27/05/2000

Cuarto parcial: Laboratorio Ponderación 20%
Tres prueba de laboratorio de 30, 30 y 40
respectivamente. Fecha: Planificada con los alumnos.

BIBLIOGRAFÍA:

Título del Libro. "Diseño y análisis de algoritmos funcionales e imperativos".
Autor Javier Galve. Juan C. González, Ángel Sánchez y J. Ángel Velázquez.
Editorial Serie Paradigma.

Título del Libro. Serie Paradigma. "Desarrollo de algoritmos y sus aplicaciones en basic, pascal. cobol y c"
Autor Guillermo Correa Uribe.
Editorial McGrawHill.

Título del Libro. "Fundamentos de Algoritmia".
Autor Brassard y P. Bratley.
Editorial Prentice Hall.