


<p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p>  <p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p> <p style="text-align: center;">PROGRAMA ANALÍTICO DE LA UNIDAD DE APRENDIZAJE:</p> <p style="text-align: center;">LENGUAJES DE PROGRAMACIÓN II</p>	DES:	INGENIERÍA
	Programa Educativo	Ingeniería en Sistemas Computacionales en Hardware
	Tipo de materia (Obli/Opta):	Obligatoria
	Clave de la materia:	415
	Semestre:	4
	Área en plan de estudios (G, E):	Ciencias de la ingeniería
	Total de horas por semana:	5
	<i>Teoría: Presencial o Virtual</i>	3
	<i>Laboratorio o Taller:</i>	2
	<i>Prácticas:</i>	0
	<i>Trabajo extra-clase:</i>	0
	Créditos Totales:	5
	Total de horas semestre (x 16 sem):	80
	Fecha de actualización:	Enero 2023
<i>Prerrequisito (s):</i>	Lenguajes de programación I, (315)	
<i>Realizado por:</i>	Comité de Rediseño Curricular	

DESCRIPCIÓN:

El curso aporta al estudiante las herramientas, técnicas y métodos matemáticos que son fundamentales para profundizar en temas tales como: Inteligencia Artificial, Sistemas Expertos, Bases de Datos, Análisis de Algoritmos, Lenguajes de Programación, entre otros.

DOMINIOS (Se toman de las competencias)	OBJETOS DE ESTUDIO (Contenidos necesarios para desarrollar cada uno de los dominios)	RESULTADOS DE APRENDIZAJE (Se plantean de los dominios y contenidos)	METODOLOGÍA (Estrategias, secuencias, recursos didácticos)	EVIDENCIAS (Productos tangibles que permiten valorar los resultados de aprendizaje)
<p>Específicas.</p> <p>Competencia:</p> <p>Sistemas informáticos y computación</p> <p>Descripción:</p>	<p>UNIDAD I. ASIGNACIÓN DINÁMICA DE MEMORIA</p> <p>1.1 Gestión dinámica de la memoria. 1.2 Función de asignación de memoria malloc(). 1.3 La función free(). 1.4 Funciones de asignación calloc() y realloc(). 1.5 Asignación dinámica para arrays. 1.6 Arrays dinámicos.</p>	<p>Identifica el manejo de memoria dinámica. Aplica los conocimientos adquiridos para implementar programas de computadora que realicen un eficiente uso de la memoria.</p>	<p>Lecturas comentadas Exposiciones por parte del maestro Materiales Gráficos: artículos, libros, etc. Cañón Pizarrón Ejercicios</p>	<p>Tareas de investigación Prácticas utilizando el programa Dev C++ Exposiciones del maestro</p>
<p>Aplica el conocimiento, metodologías, procesos y técnicas, para el análisis, diseño, modelado y desarrollo de</p>	<p>UNIDAD II. ARCHIVOS</p> <p>2.1. Apuntador FILE. 2.2. Apertura de un archivo. 2.2.1. Modos de apertura de un archivo. 2.2.2. NULL y EOF. 2.2.3. Cierre de archivos. 2.2.4. Volcado del buffer:</p>	<p>Describe y utiliza las funciones para el manejo de archivos. Realizará programas que ejemplifiquen su uso.</p>	<p>Lecturas comentadas Exposiciones por parte del maestro Materiales Gráficos: artículos, libros, etc. Cañón Pizarrón Ejercicios</p>	<p>Prácticas utilizando el programa Dev C++ Exposiciones del maestro</p>

<p>sistemas informáticos y de cómputo</p> <p>Dominio: Diseña y aplica algoritmos, estructuras y representación de datos para soluciones computacionales. Aplica las bases de los lenguajes de programación para generar aplicaciones óptimas. Aplica los tópicos de paradigmas de programación: estructurado, orientado a objetos. Lógico, funcional, entre otros para la creación de aplicaciones óptimas.</p>	<p>fflush().</p> <p>2.3. Funciones de entrada / salida para archivos.</p> <p>2.3.1. Funciones putc() y fputc().</p> <p>2.3.2. Funciones getc() y fgetc().</p> <p>2.3.3. Funciones fputs() y fgets().</p> <p>2.3.4. Funciones fprintf() y fscanf().</p> <p>2.3.5. Función feof().</p> <p>2.3.6. Función rewind().</p> <p>2.4. Archivos binarios en C.</p> <p>2.4.1. Función de salida fwrite().</p> <p>2.4.2. Función de lectura fread().</p> <p>2.5. Funciones para acceso aleatorio.</p> <p>2.5.1. Función fseek().</p> <p>2.5.2. Función ftell().</p> <p>2.5.3. Cambio de posición: fgetpos() y fsetpos().</p> <p>2.6. Datos externos al programa con argumentos en main().</p>			
	<p>UNIDAD III. ORGANIZACIÓN DE DATOS EN UN ARCHIVO</p> <p>3.1. Registros.</p> <p>3.1.1. Clave.</p> <p>3.1.2. Registro físico (bloque).</p> <p>3.2. Organización de archivos.</p> <p>3.2.1. Organización secuencial.</p> <p>3.2.2. Organización directa.</p> <p>3.3. Archivos con función de direccionamiento Hash.</p> <p>3.3.1. Funciones Hash.</p> <p>3.3.2. Características de un archivo con direccionamiento hash.</p> <p>3.4. Archivos secuenciales indexados.</p> <p>3.4.1. Partes de un archivo secuencial indexado.</p> <p>3.4.2. Proceso de un archivo secuencial indexado.</p> <p>3.5. Ordenación de archivos: ordenación externa.</p> <p>3.5.1. Fusión de archivos.</p>	<p>Describe y desarrolla los conceptos de la organización de datos en un archivo. Aplicará los conocimientos adquiridos para desarrollar programas de computadora que realicen eficientemente el manejo de datos dentro de un archivo.</p>	<p>Lecturas comentadas</p> <p>Exposiciones por parte del maestro</p> <p>Materiales Gráficos: artículos, libros, etc.</p> <p>Cañón</p> <p>Pizarrón</p> <p>Ejercicios</p>	<p>Practicas utilizando el programa Dev C++</p> <p>Exposiciones del maestro</p>

	<p>3.6. Clasificación por mezcla directa. 3.6.1. Codificación del algoritmo mezcla directa.</p>			
	<p>UNIDAD IV. TIPOS ABSTRACTOS DE DATOS (TAD /OBJETOS). (CONCEPTUAL).</p> <p>4.1. El papel (el rol) de la abstracción. 4.1.1. La abstracción como un proceso natural mental. 4.1.2. Historia de la abstracción del software. 4.1.3. Procedimientos. 4.1.4. Módulos. 4.1.5. Tipos abstractos de datos. 4.1.6. Objetos. 4.2. Tipos de datos. 4.3. Abstracción en lenguajes de programación. 4.3.1. Abstracciones de control. 4.3.2. Abstracciones de datos. 4.4. Tipos abstractos de datos. 4.4.1. Ventajas de los tipos abstractos de datos (TAD). 4.4.2. Implementación de los TAD. 4.5. Especificación de un TAD. 4.5.1. Especificación informal de un TAD. 4.5.2. Especificación formal de un TAD. 4.6. Ejemplo práctico: TAD conjunto. 4.7. Orientación a objetos. 4.7.1. Abstracción. 4.7.2. Encapsulación. 4.7.3. Modularidad. 4.7.4. Jerarquía. 4.7.5. Polimorfismo. 4.7.6. Otras propiedades. 4.7.7. Lenguajes de programación orientados a objetos. 4.8. Reutilización de software. 4.9. Desarrollo tradicional Orientado a objetos.</p>	<p>Describe y los conceptos de abstracción, tipos de datos, tipos de datos abstractos y conocerá los fundamentos del modelo orientado a objetos.</p>	<p>Lecturas comentadas Exposiciones por parte del maestro Materiales Gráficos: artículos, libros, etc. Cañón Pizarrón Ejercicios</p>	<p>Exposiciones del maestro</p>

FUENTES DE INFORMACIÓN (Bibliografía, direcciones electrónicas)	EVALUACIÓN DE LOS APRENDIZAJES (Criterios, ponderación e instrumentos)
<ol style="list-style-type: none"> 1. Philippakis Andreas S. y Kazmier Leonar J. (1998). COBOL Estructurado. McGraw Hill. 2. García Fernando, Cerro Almudena y Díez Perla J. Manuel. (1989). Programación COBOL: Algoritmos Estructurados. McGraw Hill. 3. Newcomer Lawrence R. (1991). Programación en COBOL estructurado. McGraw-Hill Interamericana de México. 	<p>Se toma en cuenta para integrar calificaciones parciales:</p> <ul style="list-style-type: none"> • Discusión Individual y por equipo, tareas y prácticas, lo cual otorga un valor del 20% • 3 Exámenes parciales escritos donde se evalúan conocimientos, comprensión y aplicación con un valor de 80% cada uno. <p>La acreditación del curso se integra por promedio de las 3 calificaciones parciales.</p>

CRONOGRAMA

Objetos de estudio	Semanas																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Unidad I: Tópicos selectos de circuitos.																	
Unidad II: Archivos																	
Unidad III: Organización de datos en un archivo																	
Unidad IV: Tipos abstractos de datos (TAD/OBJETOS). Conceptual																	