UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA **PROGRAMA ANALÍTICO DE LA UNIDAD DE APRENDIZAJE:** **LENGUAJES DE PROGRAMACIÓN I**	**DES:**	**INGENIERÍA**
	Programa Educativo	Ingeniería en Sistemas Computacionales en Hardware
	Tipo de materia (Obli/Opta):	Obligatoria
	Clave de la materia:	315
	Semestre:	3
	Área en plan de estudios (G, E):	Ciencias de la ingeniería
	Total de horas por semana:	5
	Teoría: Presencial o Virtual	3
	Laboratorio o Taller:	0
	Prácticas:	2
	Trabajo extra-clase:	0
	Créditos Totales:	5
	Total de horas semestre (x 16 sem):	80
	Fecha de actualización:	Enero 2023
	Prerrequisito (s):	215 Diseño de Algoritmos
	Realizado por:	Comité de Rediseño Curricular

DESCRIPCIÓN:

Este curso aporta las bases para que el estudiante aplique los dominios conceptuales y procedimentales adquiridos en el programa de diseño de algoritmos; esta materia se imparte en el lenguaje de programación C y C++ buscando que el alumno sea capaz de identificar los beneficios de este lenguaje así como de cualquier otro, siendo capaz de realizar un programa o sistemas estructurados y así pueda transferir sus dominios a otro(s) lenguaje(s) requiriendo exclusivamente la sintaxis del mismo.

Al final del curso el estudiante:

- Está capacitado para utilizar las estructuras de control (selección y repetición)
- Crea y manipula funciones, arreglos y estructuras de datos
- Utiliza apuntadores como variables dinámicas.

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)
Especificas: **Competencia:** ***Sistemas Informáticos y Computación*** ***Descripción:*** Aplica el conocimiento, metodologías, procesos y técnicas, para el análisis, diseño, modelado y desarrollo de sistemas informáticos y de cómputo.	**UNIDAD I.** ***INTRODUCCIÓN A LA PROGRAMACIÓN ESTRUCTURADA.*** 1.1. Conceptos generales. 1.1.1. Variables, Constantes. 1.1.2. Instrucciones de Entrada / Salida. 1.1.3. Editor. 1.2. Aritmética en C. 1.3. Formatos de Control. 1.4. Archivos de inclusión. 1.5. Programación Secuencial.	Define y escribe programas simples en lenguaje C, utilizando para ello los enunciados básicos de entrada y salida. Además, identificará los tipos fundamentales de datos comprendiendo para ello los conceptos de memoria de la computadora. Será capaz de utilizar los operadores aritméticos para cálculos sencillos.	• Lectura crítica. • Auto aprendizaje (búsqueda y análisis de información). • Clase Interactiva Maestro-Alumno. • Técnicas de enseñanza demostrativa. • Aprendizaje basado en prácticas de laboratorio (ABPL). • Recursos tecnológicos.	• Tareas de Investigación. • Exposiciones. • Prácticas de Laboratorio. • Actividad integradora. • Examen.

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.	**UNIDAD II.** ***DESARROLLO DE PROGRAMAS ESTRUCTURADOS*** 2.1. Programación Condicional. 2.1.1. Instrucciones de Condición. 2.1.1.1. Condicional Simple If-Else. 2.1.1.2. Condicional anidada. 2.1.1.3. Condicional por casos Switch–Case. 2.2. Ciclos. 2.2.1. Tipos de ciclos y sintaxis. 2.2.1.1. Ciclos For. 2.2.1.2. Ciclos While. 2.2.1.3. Ciclos Do-While. 2.2.2. Ciclos anidados. 2.2.3. Demostración de ciclos infinitos.	Expresa la importancia de la programación estructurada y será capaz de desarrollar programas que incluyan enunciados de toma de decisiones (IF - ELSE), selección múltiple (SWITCH) y de repetición controlada por contador y por centinela (FOR, WHILE y Do-WHILE). Además de tener la capacidad para utilizar los operadores lógicos, incrementales y decreméntales.	• Lectura crítica. • Auto aprendizaje (búsqueda y análisis de información). • Clase Interactiva Maestro-Alumno. • Técnicas de enseñanza demostrativa. • Aprendizaje basado en prácticas de laboratorio (ABPL). • Recursos tecnológicos.	• Tareas de Investigación. • Exposiciones. • Prácticas de Laboratorio. • Actividad integradora. • Examen.
	UNIDAD III. ***FUNCIONES*** 3.1. Conceptos generales de funciones. 3.2. Funciones matemáticas de la biblioteca estándar de C. 3.3. Definiciones de función. 3.4. Prototipo de funciones. 3.5. Cómo llamar funciones: llamada por valor y llamada por referencia. 3.6. Recursividad y ejemplos.	Identifica cómo construir programas en forma modular a partir de funciones. Utiliza funciones comunes de C, creará funciones nuevas y comprende los mecanismos utilizados para pasar información a funciones.	• Lectura crítica. • Auto aprendizaje (búsqueda y análisis de información). • Clase Interactiva Maestro-Alumno. • Técnicas de enseñanza demostrativa. • Aprendizaje basado en prácticas de laboratorio (ABPL) • Recursos tecnológicos	• Tareas de Investigación • Exposiciones • Prácticas de Laboratorio • Actividad integradora. • Examen

	UNIDAD IV: ARREGLOS 4.1.1. Conceptos generales. 4.1.2. Tipos de arreglos. 4.1.2.1. Arreglos Unidimensionales. 4.1.2.2. Arreglos Bidimensionales. 4.1.3. Paso de arreglos a funciones. 4.1.4. Ejemplos utilizando arreglos.	Describe los conceptos de arreglos de datos, comprendiendo el uso de los arreglos para almacenar, ordenar y buscar listas y tablas de valores. Será capaz de operar arreglos en funciones.	• Lectura critica • Auto aprendizaje (búsqueda y análisis de información) • Clase Interactiva Maestro-Alumno • Técnicas de enseñanza demostrativa • Aprendizaje basado en prácticas de laboratorio (ABPL) • Recursos tecnológicos	• Tareas de Investigación • Exposiciones • Prácticas de Laboratorio • Actividad integradora. • Examen
	UNIDAD V: ESTRUCTURAS Y UNIONES 5.1. Estructuras. 5.1.1. Declaración de una estructura. 5.1.2. Definición de variables de estructuras. 5.1.3. Uso de estructuras en asignaciones. 5.1.4. Inicialización de una declaración de estructuras. 5.1.5. El tamaño de una estructura. 5.2. Acceso a Estructuras. 5.2.1. Almacenamiento de información en estructuras. 5.2.2. Lectura de información de una estructura. 5.2.3. Recuperación de información de una estructura. 5.3. Sinónimo de un tipo de dato: typedef. 5.4. Estructuras anidadas. 5.4.1. Ejemplos de estructuras anidadas.	El alumno será capaz de reconocer los conceptos de estructuras y uniones dentro de programas estructurados.	• Lectura critica • Auto aprendizaje (búsqueda y análisis de información) • Clase Interactiva Maestro-Alumno • Técnicas de enseñanza demostrativa • Aprendizaje basado en prácticas de laboratorio (ABPL) • Recursos tecnológicos	• Tareas de Investigación • Exposiciones • Prácticas de Laboratorio • Actividad integradora. • Examen

	5.5. Arrays de estructuras. 5.6. Arrays como miembros. 5.7. Utilización de estructuras como parámetros. 5.8. Uniones. 5.9. Tamaño de estructuras y uniones.			
	UNIDAD VI: APUNTADORES 6.1 Definición. 6.2 Declaración e inicialización de variables de apuntadores 6.2.1. Declaración de apuntadores. 6.2.2. Inicialización de apuntadores. 6.2.3. Apuntadores y verificación de tipos. 6.2.4. Apuntadores NULL y void. 6.3 Operaciones de apuntador. 6.3.1. Apuntadores a arrays. 6.3.2. Apuntadores a cadenas. 6.3.3. Apuntadores a estructuras. 6.4 Expresiones y aritmética de apuntadores.	El alumno será capaz de identificar los apuntadores, describirá la relación entre apuntadores, arreglos, estructuras y cadenas. Desarrolla la utilización de apuntadores a funciones y será capaz de declarar y utilizar arreglos de cadenas.	• Lectura critica • Auto aprendizaje (búsqueda y análisis de información) • Clase Interactiva Maestro-Alumno • Técnicas de enseñanza demostrativa • Aprendizaje basado en prácticas de laboratorio (ABPL) • Recursos tecnológicos	• Tareas de Investigación • Exposiciones • Prácticas de Laboratorio • Actividad integradora y/o Proyecto Final • Examen

FUENTES DE INFORMACIÓN (Bibliografía, direcciones electrónicas)	**EVALUACIÓN DE LOS APRENDIZAJES** (Criterios, ponderación e instrumentos)

1. Paul J. Deitel, Harvey M. Deitel. (2004). Como programar en C/C++ y Java. Ed 4. Pearson.
2. Herb Schildt. (2008). C++ Programming Cookbook. Mc Graw Hill
3. Luis Joyanes Aguilar.(2006). Programación en C++ (Algoritmos, estructuras de datos y objetos).Mc Graw Hill
4. Luis Joyanes Aguilar y Ignacio Zahonero Martínez. (2001). Programación en C Metodología, Estructuras de Datos y Objetos. McGraw Hill.
	INSTRUMENTOS:
	• ***Actividades:***
	  • Tareas de Investigación
	  • Exposiciones
	  • Prácticas de Laboratorio y/o actividad integradora y/o Proyecto.
	• ***Examen***
	
	CRITERIOS DE DESEMPEÑO:
	Las actividades deberán estar completas y entregadas en tiempo y forma.
	
	Exámenes:
	Se realizan 3 exámenes durante el semestre y las fechas se establecerán por la secretaría académica.
	
	Se toma en cuenta para integrar calificaciones parciales:
	
	Exámenes 70%
	Actividades 30%
	
	Observaciones
	
	• Las actividades no realizadas en tiempo y forma se califican con cero.
	
	• Para acreditar el curso se deberá tener calificación aprobatoria tanto en exámenes y actividades.
	
	• **Exposición:** presentadas en orden lógico:
	  1. Introducción resaltando el objetivo a alcanzar
	  2. Desarrollo temático, responder preguntas y aclarar dudas
	  3. Concluir.
	
	• **Actividades**
	
	Toda actividad complementaria al curso se podrá llevar a cabo en forma individual o por equipo según amerite el tema. Estos se reciben únicamente en tiempo y forma previamente establecidos.
	
	• **Prácticas de Laboratorio**:
	
	Ajustarse al formato que se utiliza en el laboratorio.

CRONOGRAMA

Objetos de estudio	Semanas															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
UNIDAD I: INTRODUCCIÓN A LA PROGRAMACIÓN ESTRUCTURADA																
UNIDAD II: DESARROLLO DE PROGRAMAS ESTRUCTURADOS																
UNIDAD III: FUNCIONES																
UNIDAD IV: ARREGLOS																
UNIDAD V: ESTRUCTURAS Y UNIONES																
UNIDAD VI: APUNTADORES																