

1.- Identificación de la Unidad de Aprendizaje					
Nombre de la Unidad de Aprendizaje					
PROGRAMACION ESTRUCTURADA					
Clave de la UA	Modalidad de la UA	Tipo de UA		Valor de créditos	Área de formación
F0092	<i>Presencial</i>	<i>Curso-Taller</i>		6	Básica Particular Obligatoria
Hora semana		Horas teoría/semestre	Horas práctica/semestre	Total de horas:	Seriación
4		40	40	80	No tiene
Departamento			Academia		
<i>Ciencias de la Información y Desarrollo Tecnológico</i>			Ingeniería de Software		
Presentación					
En esta unidad de aprendizaje el alumno comprenderá los conceptos de programación estructurada así como sus componentes y su aplicación y uso en el lenguaje de programación C, con este aplicara e implementara en códigos los conceptos que le permitan soluciones y aplicaciones de software, al escribir en código sus algoritmos.					
Tipos de saberes					
Saber (Conocimientos)	Saber hacer (Habilidades)		Saber ser (Actitudes y valores)		
Conceptos de algoritmos, escribir pseudocódigo, diseñar problemas, conocer la programación estructurada, conocer un ambiente integrado de programación, el proceso de traducción de código humano a máquina, localizar errores de código, conocer técnicas de depuración, conocer los elementos de programa: variable, tipos de datos funciones, métodos y librerías, conocer las estructuras de control de lenguaje de programación C y conocer los tipos de Entrada y salida existentes..	Operadores aritméticos y su implementación en C. Codificar, compilar y ejecutar un programa sencillo. Codificar, compilar y ejecutar en un programa un algoritmo o pseudocódigo. Seleccionar una Interfaz de programación, saber traducir un pseudocódigo a lenguaje de programación C. Usar y declarar variables, tipos de Datos en Lenguaje C, así como saber usar las estructuras de Ciclo y Control y el manejo de entrada y salida a archivos en funciones de C.		Receptivo Analítico Propositivo Participativo Compromiso en su desempeño Honesto		
Competencia genérica			Competencias profesionales		

<p>Escribe algoritmos para solución de problemas y construye aplicaciones haciendo uso del lenguaje de Programación C.</p>	<ul style="list-style-type: none"> • Memoriza el concepto de computadora para posteriormente referirlo en el contexto de las ciencias computacionales. • Explica que es un algoritmo, los tipos de algoritmos y describe tareas usando algoritmos. • Distingue los paradigmas de programación, explica que son su evolución y su uso. • Identifica las diferentes etapas de la programación para seguirlas en la práctica diaria de la programación. • Relaciona las fases de la ejecución de un programa para identificar problemas en la ejecución de estos. • Argumenta las características de los lenguajes de programación para clasificarlos en alto y bajo nivel. • Explica los tipos de variables y reconoce sus límites. • Utiliza constructos de ciclos y de alternación para la construcción de aplicaciones de software y en la implementación de soluciones de problemas computables. • Aplica los conocimientos adquiridos para analizar, diseñar e implementar programas usando el lenguaje C. • Utiliza la construcción de archivos para el almacenamiento y la persistencia de datos.
<p>Saberes previos del alumno</p>	
<p>Comprensión oral y escrita, aritmética básica.</p>	
<p>Perfil de egreso al que se abona</p>	
<p>Utiliza modelos matemáticos, aplica algoritmos y principios de las ciencias computacionales en el modelado y diseño de sistemas de información, manejo del lenguaje de programación imperativo</p>	
<p>Perfil deseable del docente</p>	
<p>Conocimientos:</p> <ul style="list-style-type: none"> • Arquitecturas de computadoras • Aritmética modular • Sistemas de numeración • Programación en lenguajes de bajo nivel 	
<p>Habilidades:</p> <ul style="list-style-type: none"> • Buena comunicación oral y escrita • Planificar el proceso de enseñanza 	

- Interactuar de manera empática con los alumnos

2.- Contenidos temáticos	
Contenido	
MODULO 1. INTRODUCCION A LA PROGRAMACION	
1.1 Historia de las computadoras	<ul style="list-style-type: none"> • Paradigmas de programación • Lenguajes de Programación • Clasificación de lenguajes de programación • Traductores: Compiladores vs Interpretes
1.2 Desarrollo de Algoritmos usando pseudocódigo.	<ul style="list-style-type: none"> • Método de Ingeniería de Software • Enfoques para la solución de problemas procedurales: asignaciones, condiciones y ciclos • Problemas de fórmulas clásicos
1.3 Estructuras básicas de programa y ambientes para el desarrollo de Aplicaciones.	<ul style="list-style-type: none"> • Estructura básica de un programa • Documentación y prácticas estándar de programación • Diferentes ambientes para el desarrollo de aplicaciones (IDEs) • Edición, compilación y ejecución en IDE para lenguaje C. • Como depurar errores de programas en IDE para Lenguaje C.
1.4 Básicos de Entrada y Salida	<ul style="list-style-type: none"> • Formateo de datos y salida a consola • Entrada por consola • Formateo de salida
1.5 Variables y Expresiones	<ul style="list-style-type: none"> • Declaración de variables y tipos básicos de datos (ej. int, long, float) • Expresiones aritméticas, precedencia de operadores y asociaciones • Operadores de asignación abreviados • Biblioteca de funciones • Solución de problemas usando variables y expresiones
MODULO 2. ESTRUCTURAS DE CONTROL DEL FLUJO DE EJECUCION	
2.1 Estructura de alternación/condición	<ul style="list-style-type: none"> • Expresiones booleanas • Estatutos de alternación condicional simple (if) • Estatutos de alternación condicional compuestos (if/else) • Estatutos multi-camino (switch/case) • Estructuras de alternación anidadas. • Solución de problemas usando estructuras de alternación
2.2 Estructuras de ciclos	<ul style="list-style-type: none"> • Variables para control en ciclos, inicialización, prueba y modificaciones • Ciclo con condición al inicio (ej. while) • Ciclo con condición al final (ej. do/while) • Ciclo por conteo (ej. for) • Estructuras de ciclo anidadas • Solución de problemas utilizando estructuras de ciclo

MODULO 3. FLUJOS DE ENTRADA Y SALIDA

3.1 Estructuras de datos simples

- Arreglos de una dimensión
- Cadenas(strings) como arreglos
- Arreglos multidimensionales
- Registros (ej. Objetos/entidades)
- Solución de problemas usando archivos de entrada y salida

3.2 Entrada y salida usando archivos

- Leyendo entrada desde archivos
- Lecturas en ciclos
- Salida hacia archivos
- Solución de problemas usando archivos de entrada y salida

MODULO 4. Funciones o subprogramas

4.1 Estatuto de una función

- Paso de Argumentos
- Retorno de resultados
- Solución de problemas usando funciones

4.2 Recursión

- Tipos de recursión
- Paso básico y recursivo en la definición de funciones recursivas
- Comportamiento de las variables utilizadas en funciones recursivas
- Solución de problemas básicos usando recursividad

Estrategias generales para impartir la unidad de aprendizaje

Disertación tradicional, material audiovisual, material de apoyo en línea, videos en línea, escritura de programas en IDE, prácticas en laboratorio.

Módulo I

INTRODUCCION A LA PROGRAMACION

Competencia Específica

- Memoriza el concepto de algoritmo, lenguaje y programa para posteriormente referirlo en el contexto de las ciencias computacionales.
- Distingue los paradigmas de programación existentes para conocer la aplicación de cada una en problemas específicos, conoce la historia de las computadoras y los tipos de lenguajes de programación existentes.

Tipos de saberes

Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Concepto de Algoritmos, pseudocódigo y tipos lenguajes, programación estructurada y componentes de una computadora.	Escribir algoritmos usando el lenguaje C para programación estructurada C	Receptivo Analítico Propositivo Participativo

		Compromiso en su desempeño
Módulo II		
ESTRUCTURAS DE CONTROL DEL FLUJO DE EJECUCION		
Competencia Específica		
<ul style="list-style-type: none"> • Aplica los cambios de base para realizar operaciones aritméticas en los diferentes sistemas de representación. • Identifica los diferentes formatos de archivos para asociar el correcto según la información a representar. 		
Tipos de saberes		
Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Sistemas de numeración, cambio de base, aritmética modular, codificación y formatos de archivos.	Operaciones aritméticas en diferentes sistemas de numeración, obtener el módulo de un número.	Receptivo Analítico Propositivo Participativo Compromiso en su desempeño
Módulo III		
FLUJOS DE ENTRADA Y SALIDA		
Competencia Específica		
<ul style="list-style-type: none"> • Identifica las diferentes etapas de la programación para seguirlas en la práctica diaria de la programación. • Relaciona las fases de la ejecución de un programa para identificar problemas en la ejecución de los mismos. • Argumenta las características de los lenguajes de programación para clasificarlos en alto y bajo nivel. 		
Tipos de saberes		
Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Estructura de un programa en memoria, fases de la compilación y ejecución, características de los lenguajes de alto y bajo nivel.	Codificar, compilar y ejecutar un programa sencillo.	Receptivo Analítico Propositivo Participativo Compromiso en su desempeño Honesto
Módulo IV		

FUNCIONES O SUBPROGRAMAS		
Competencia Específica		
<ul style="list-style-type: none"> • Explica el funcionamiento del lenguaje ensamblador para mostrar su ejecución en los registros físicos del CPU. • Señala el ciclo de arranque del sistema operativo para identificar problemas asociados con el mismo. 		
Tipos de saberes		
Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Características del lenguaje ensamblador, partes de los CPU's modernos, características del sistema operativo.	Codificar, compilar y ejecutar un programa rudimentario en lenguaje ensamblador. Observar la ejecución de un programa en el CPU.	Receptivo Analítico Propositivo Participativo Compromiso en su desempeño Honesto
Bibliografía básica		
<ul style="list-style-type: none"> • Sznajdleder, Pablo.(2017). Programación Estructurada a fondo. Editorial alfaomega. ISBN: 978-987-3832-28-4. • Ceballos-Sierra, F. (2019). C/C++. Curso de programación. Editorial Ra-Ma. ISBN: 978-84-9964-812-5 • VozMediano, A.M. (2017). Aprender a programar en C. De 0 a 99 en un solo libro. Editorial Independently published. ISBN-13: 978-1521018958, ISBN-10: 1521018952 		
Bibliografía complementaria		
<ul style="list-style-type: none"> • Méndez-Girón, Alejandra. (2013). Diseño de algoritmos y su programación en C. Editorial Alfaomega, ISBN 9786077076018. • SFuenlabrada-Velazquez, S. Miranda, E. (2015). Manejo de técnicas de programación. Enfoque por competencias (eBook). Editorial Pearson Education. ISBN: 9786073232432 		
3.-Evaluación		
Criterios de Evaluación (% por criterio)		
Exámenes	50%	
Practicas	50%	
4.- Participantes en la elaboración		
Código	Nombre	
9528377	Dr. Manuel Corona Pérez	
2021935	Dra. Graciela Villanueva Álvarez	
5.- Fecha de elaboración		
Elaboración: Enero 2019		
Actualización: Febrero 2021.		9814671 Marisela Mireles Mercado