|  |
| --- |
| **OBJETIVOS:**  **Objetivo General:**  Al final del curso el alumno será capaz de diseñar algoritmos eficaces bajo el paradigma de la programación estructurada para dar solución a problemas computacionales.  **Objetivos específicos:**   1. Aplicar los conceptos del diseño de algoritmos en la programación estructurada. 2. Traducir algoritmos a un lenguaje imperativo de programación usando la sintaxis correcta. 3. Probar la eficacia de los programas al ejecutarlos en una computadora. 4. Aplicar el principio de divide y vencerás en el contexto del diseño e implementación de algoritmos. 5. Aplicar el principio de diseño de arriba abajo (top-down) para el diseño e implementación de algoritmos.   **CONTENIDO SINTÉTICO:**   1. Estructura de control secuencial.    1. Identificadores, tipos de datos, variables.    2. Asignación, expresiones y operadores.    3. Entrada de datos desde el teclado y salida hacia la pantalla. 2. Estructuras de control selectiva    1. Sencilla.    2. Anidada.    3. Múltiple. 3. Estructura de control iterativa.    1. Sencilla con número determinado de iteraciones.    2. Con número indeterminado de iteraciones.    3. Control previo.    4. Control posterior.    5. Iteración anidada. 4. Arreglos    1. Declaración e inicialización de arreglos.    2. Acceso a los elementos de un arreglo.    3. Recorrido de arreglos.    4. Arreglos de varias dimensiones.    5. Operaciones con cadenas de caracteres. 5. Registros    1. Importancia y alcance de los registros.    2. Definición y acceso de los campos de un registro.    3. Registros anidados.    4. Arreglos de registros. 6. Manejo de referencias.    1. Concepto de referencia.    2. Inicialización de referencias.    3. Referenciación de variables.    4. Desreferenciación de variables. 7. Diseño modular    1. La importancia del diseño modular.    2. Diseño Top-Down contra Bottom-Up.    3. Subrutinas y funciones.    4. Paso de parámetros por valor y por referencia.    5. Uso de librerías. 8. Archivos    1. Definición y tipos de archivos.    2. Operaciones sobre archivos.    3. Archivos con acceso secuencial y aleatorio.   **MODALIDADES DE CONDUCCIÓN DEL PROCESO DE ENSEÑANZA-APRENDIZAJE:**  Clase teórico-práctica a cargo del profesor con participación activa del alumno y clase práctica en laboratorio.  El profesor diseñará experiencias de aprendizaje por problemas con nivel de complejidad incremental, tanto en el aula como en el laboratorio. El alumno analizará los problemas planteados y aplicará los conceptos aprendidos durante el curso para la solución de los problemas en clase y en las prácticas de laboratorio.  Las habilidades transversales que deberá adquirir el alumno asociadas a esta UEA son las siguientes:  (Ht0) Lenguaje disciplinar: Aprenderá conceptos de un lenguaje de programación estructurada.  (Ht1) Autoaprendizaje.  (Ht4) Comprender perfectamente los textos técnicos en español: El profesor promoverá que el alumno investigue cómo escribir algunos de los programas vistos en clase en cualquier otro lenguaje de interés para el alumno.  (Ht6) Comprender textos técnicos en inglés: Leerá un texto técnico en inglés y lo explicará en español. El texto deberá ser breve (media cuartilla).  Las habilidades disciplinares que deberá adquirir el alumno asociadas a esta UEA son:  (H1) Abstracción de sistemas.  (H2) Resolver problemas mediante algoritmos: Analizará los problemas propuestos que involucran las estructuras básicas de control y los resolverá mediante un algoritmo que implementará en algún lenguaje de programación.  Las actitudes que deberá mostrar el alumno son:  (A2) Perseverancia para resolver problemas mediante la programación estructurada.  (A3) Disciplina para aplicar los conocimientos adquiridos.  **MODALIDADES DE EVALUACIÓN:**  **Evaluación Global:**  Se ponderarán las siguientes actividades a criterio del profesor:   * Tareas individuales, que incluyen prácticas de laboratorio. * Evaluaciones periódicas. * Participación en el proceso de resolución de problemas, tanto en las sesiones teóricas como prácticas. * Evaluación terminal. * Evaluación de la capacidad del alumno para extraer información de un texto técnico en inglés. * Traducción de un programa de un lenguaje de programación a otro.   **Evaluación de Recuperación:**  El alumno deberá presentar una evaluación que contemple todos los contenidos de la unidad de enseñanza aprendizaje. A criterio del profesor, se podrá solicitar también un proyecto de cómputo.  No requiere inscripción previa a la UEA.  **BIBLIOGRAFÍA NECESARIA O RECOMENDABLE:**   1. Antonakos, J. L. y Mansfield, K. C. *Programación estructurada en C*. Prentice Hall, México, 2000. 2. Dale, N. Y Weems, C. *Programming and problem solving with C++*. John and Bartlett, Massachusetts, 2005. 3. Joyanes, A. L. *Fundamentos de programación*. McGraw Hill, 4ª ed., España, 2008. 4. Kernighan, B. y Ritchie, D. *El lenguaje de programación C*. Prentice-Hall, 2ª Edición, México, 1991. 5. Levine, G. *Programación estructurada y fundamentos de programación.* McGraw Hill, 2da Edición, México, 1990. 6. López, L. *Programación estructurada: un enfoque algorítmico.* Alfaomega, 2ª ed, México, 2008. 7. Peñaloza Romero, E. *Fundamentos de programación C/C++*. Alfaomega, 4ª ed., México, 2004. |