



Programación Básica

| Programa sintético | | | | |
|----------------------|---|--|------------------------------------|----------|
| Programación Básica | | | | |
| Datos básicos | | | | |
| Semestre | Horas de teoría | Horas de práctica | Horas trabajo adicional estudiante | Créditos |
| 1 | 3 | 2 | 3 | 8 |
| Objetivos | Identificar y aplicar los conceptos básicos de programación estructurada en un lenguaje de alto nivel. Al final del curso, el alumno deberá ser capaz de diseñar, implementar, y depurar algoritmos sencillos en lenguaje C/C++ o Python. | | | |
| Temario | Unidades | Contenidos | | |
| | 1. Conceptos básicos de programación | 1.1.- Estructura básica de un programa 1.2.- Salida a consola mediante cout 1.3.- Compilación y ejecución de un programa 1.4.- Variables y asignación 1.5.- Expresiones aritméticas y jerarquía de operadores 1.6.- Entrada de datos mediante cin 1.7.- Almacenamiento de variables en memoria 1.8.- Apuntadores y operadores de referenciación y dereferenciación 1.9.- Aritmética de apuntadores 1.10.- Ejemplos de programas sencillos | | |
| | 2. Estructuras de decisión | 2.1.- Expresiones booleanas y operadores de comparación 2.2.- Operadores booleanos y el tipo bool 2.3.- Instrucción if...else 2.4.- Instrucciones if...else anidadas 2.5.- Instrucción switch 2.6.- Ejemplos de programas | | |
| | 3. Estructuras de iteración | 3.1.- Motivación para el uso de ciclos 3.2.- Instrucción while 3.3.- Instrucción do...while 3.4.- Instrucción for 3.5.- Instrucciones break y continue 3.6.- Ejemplos de programas | | |
| | 4. Funciones y programación estructurada | 4.1.- Ejemplos de funciones de librería: la librería math.h 4.2.- Definición de funciones y paso de parámetros por valor 4.3.- Paso de parámetros por apuntador 4.4.- Paso de parámetros por referencia | | |



| Programa sintético | | |
|--|--|---|
| | <p>4.5.- Funciones recursivas</p> <p>4.6.- Programación estructurada: motivación y recomendaciones</p> <p>4.7.- Creación de librerías: archivos de encabezado y de implementación</p> | |
| 5. Arreglos | <p>5.1.- Motivación</p> <p>5.2.- Declaración de un arreglo y acceso a sus elementos</p> <p>5.3.- Recorrido de un arreglo</p> <p>5.4.- Almacenamiento en memoria: relación entre arreglos y apuntadores</p> <p>5.5.- Ejemplos de aplicación: sumatorias, histogramas, señales</p> <p>5.6.- Arreglos bidimensionales y multidimensionales</p> <p>5.7.- Cadenas de caracteres</p> <p>5.8.- Manejo de cadenas: librería string.h</p> | |
| 6. Introducción al manejo dinámico de memoria | <p>6.1.- Asignación dinámica de memoria para una variable: operadores new y delete</p> <p>6.2.- Asignación dinámica de memoria para un arreglo</p> <p>6.3.- Consideraciones para el manejo dinámico de memoria</p> | |
| Métodos prácticos y | Métodos | Se trabajará de manera alternada la técnica expositiva con técnicas de aprendizaje basado en problemas y aprendizaje basado en proyectos para centrar el modelo en el aprendizaje del alumno. Así mismo se propiciará un uso intensivo de las tecnologías de información y comunicación para la búsqueda de información, así como la administración de un sitio web de apoyo a la clase presencial para la entrega de tareas y socialización del conocimiento. Se requiere también el uso de software de programación C/C++ o Python. |
| | Prácticas | Se sugiere la realización de una práctica por semana en las cuales el alumno deba implementar algoritmos simples, como búsquedas, métodos numéricos, estadísticas, etc. Se sugiere también desarrollar un proyecto final en el que se ataque un problema específico. |
| Mecanismos y procedimientos de evaluación | Exámenes parciales | 1 Examen teórico-práctico de las Unidades 1 y 2 con un peso máximo de 20% |
| | | 2 Examen teórico-práctico de la Unidad 3 con un peso máximo de 20% |
| | | 3 Examen teórico-práctico de la Unidad 4 con un peso máximo de 20% |



| Programa sintético | |
|--|---|
| | 4 Examen teórico-práctico de las Unidades 5 y 6 con un peso máximo de 20% |
| Examen ordinario | Proyecto final con evaluación oral y un peso máximo de 30% |
| Examen a título | Examen teórico-práctico con una duración mínima de 2 horas. |
| Examen de regularización | Examen teórico-práctico con una duración mínima de 2 horas. |
| Otros métodos y procedimientos | La asistencia y participación en clase pueden evaluarse y tener un peso no mayor al 10% de la calificación final. |
| Otras actividades académicas requeridas | |
| Bibliografía básica de referencia | C++ Como Programar. Deitel y Deitel. Prentice Hall, 2ª Edición, 1999. |
| | El Lenguaje de Programación C, Brian Kernighan, Dennis Ritchie, Ed. Prentice Hall, 2ª Edición, 1991. |
| | Métodos Numéricos para Ingenieros. S.C. Chapra, R.P. Canale. Ed. Mc Graw-Hill, 5ª Edición, 2007 |