



**ASIGNATURA: Programación II.**

**Curso: 2001/2002**

**Carácter:** Troncal

**Temporalidad:** 1<sup>er</sup> Cuatrimestre

**Créditos:** 7,5 (3T+4,5P)

**Profesor:** Andrés Santiago Martín

**Despacho:** Despacho 10

**Web:** <http://cum.unex.es/profesores/asanmar/default.htm>

**E-mail:** [asanmar@unex.es](mailto:asanmar@unex.es)

**NORMAS GENERALES:**

- Las convocatorias de los exámenes serán fijadas por la Subdirección Académica del Centro.
- Todo alumno deberá entregar obligatoriamente una ficha al profesor de la asignatura.
- En los trabajos prácticos de la asignatura el profesor establecerá, en su momento, la fecha límite de entrega de cada uno de dichos trabajos.

**OBJETIVOS GENERALES:**

- Conocer y manejar debidamente estructuras de datos tales como listas, pilas, colas, árboles y hashing.
- Conocer y manejar debidamente algoritmos de ordenación y de búsqueda.
- En las prácticas se pretende familiarizar al alumno con el manejo de los conceptos anteriores, y a su vez enseñarle a programar correctamente en programación orientada a objetos.

**METODOLOGÍA:**

- En las dos clases de teoría semanales se exponen los conceptos teóricos sobre estructuras de datos y algoritmos de ordenación y de búsqueda.
- Así mismo, están las horas de tutorías en las que los alumnos pueden consultar con el profesor la resolución de cualquier duda planteada sobre cualquier aspecto de la asignatura.
- En las clases teóricas se utilizan medios audiovisuales de divulgación informática.
- Las clases prácticas se destinan a manejar de forma práctica los conceptos vistos en teoría, y a enseñar a programar correctamente en lenguajes orientados a objetos como C++.
- En la página Web del profesor y en la copistería del centro se encuentra a disposición de los alumnos, los horarios de tutorías, temario de la asignatura, criterios de evaluación así como todo el material didáctico usado en la misma.

### CRITERIOS DE EVALUACIÓN:

- **Exámenes:**
  - Habrá un examen que constará de una parte de teoría y otra parte de problemas. La nota de este examen representará el 60% de la nota final.  
Para poder superar el examen, en cada parte del mismo (teoría y problemas) deberá obtener el alumno, al menos, el 30 % de la nota máxima que puede obtenerse en cada parte.
- **Prácticas:**
  - Habrá dos tipos de prácticas a realizar de forma individual por cada alumno.
    - Periódicamente se entregará por parte del profesor una relación de ejercicios de complejidad pequeña que el alumno deberá **entregar en el plazo de 15 días**. Cada una de estas relaciones será puntuada sobre 10, siendo la nota final de dichos ejercicios la media de todos ellos y representando dicha nota el 20 % de la nota final de prácticas. Su entrega será **obligatoria y necesaria** para poder aprobar la asignatura.
    - La práctica final consistirá en un trabajo a elegir por cada alumno de entre los propuestos por el profesor, dicho trabajo **deberá entregarse en la fecha que se indique**, y no se recogerán trabajos entregados después de esa fecha. Dicho trabajo representará el 80 % de la nota final de la parte práctica. Su entrega será **obligatoria y necesaria** para aprobar la asignatura.
  - La nota total de la parte práctica representará el 40% de la nota final.
  - Cualquier sospecha de copia sobre una práctica entregada o parte de la misma, implicará inexorablemente suspender la práctica completa, tanto a la persona que la copió como a la persona que permitió copiarla.
- **Trabajos voluntarios presentados.**
  - Dichos trabajos versarán sobre temas relacionados directamente con la asignatura, y necesitarán de la aprobación previa del profesor.
  - Estos trabajos, tras su evaluación por el profesor, podrán subir de 0,5 a 1,5 puntos que se sumarán a la nota conseguida en los apartados anteriores, siempre que en los mismos se obtenga la calificación mínima exigida.
  - La entrega de trabajos sólo se aplicará a la convocatoria de Febrero.
- La nota final será igual a:
  - Si  $(teoría \geq 4,5)$  y  $(práctica \geq 4,5)$  y  $([teoría * 0,6] + [práctica * 0,4]) \geq 5$ 
    - **nota final = (teoría \* 0,6) + (práctica \* 0,4) + nota trabajos**
  - Si no
    - **nota final = Suspenso**
  - **Fin si**
  - La nota final será igual a: **(nota examen \* 0,6) + (nota práctica \* 0,4)**
  - En ambas partes, teoría y práctica, se deberá de tener por separado una nota mayor o igual a 4,5 para poder compensar con la otra parte y poder aprobar la asignatura completa.
  - En todo caso, sólo se aprobará si la nota final es igual o superior a 5. Si un alumno tiene una parte aprobada, nota mayor o igual a 5, y tiene la otra parte suspensa (no compensable), con nota menor que 4,5, la nota que se pondrá en las actas será la nota de la parte suspensa.
  - Tanto la nota del examen teórico como la nota de la parte práctica podrán ser guardadas hasta la convocatoria de septiembre de este curso, siempre que sea superior a 5.



## **PROGRAMA TEÓRICO:**

### **TEMA 1.- GESTIÓN DE MEMORIA DINÁMICA. PUNTEROS.**

Introducción.  
El tipo puntero.  
Operaciones con punteros.  
Asignación dinámica de memoria.  
Creación dinámica de arrays.  
Punteros a estructuras.

### **TEMA 2.- INTRODUCCIÓN AL ANÁLISIS Y DISEÑO DE ALGORITMOS.**

Diseño de un Programa.  
Concepto de algoritmo.  
Descripción de algoritmos: pseudocódigo y diagramas de flujo.  
Tiempo de ejecución. La notación  $O(n)$ .  
Abstracción y Encapsulamiento.  
Tipos Abstractos de Datos.

### **TEMA 3.- INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS.**

¿Qué es Programación Orientada a Objetos?  
Un poco de historia.  
Evolución de los enfoques de programación.  
Lenguajes de POO.  
La programación como tipos abstractos de datos.  
La programación orientada a objetos.

### **TEMA 4.- ESTRUCTURAS DE DATOS LINEALES: LISTAS, PILAS Y COLAS.**

Listas enlazadas.  
Pilas y colas.  
Implementación de listas, pilas y colas.

### **TEMA 5.- ESTRUCTURAS DE DATOS AVANZADAS: ÁRBOLES Y HASHING.**

Árboles.  
Implementaciones de árboles.  
Árboles binarios.  
Implementación de árboles binarios.  
Hashing.  
Implementación de tablas de dispersión.

### **TEMA 6.- ALGORITMOS DE ORDENACIÓN Y BÚSQUEDA.**

Algoritmos de Búsqueda.  
Introducción a la Ordenación.  
Algoritmos Elementales de Ordenación.  
Ordenación por shell.  
Ordenación por mezcla.



## **PROGRAMA PRÁCTICO:**

### **TEMA 1.- CLASES Y OBJETOS.**

Clases y objetos.  
Métodos.  
Constructores.  
El parámetro this.  
Destrucción de objetos.  
Trabajando con objetos.

### **TEMA 2.- MATRICES Y CADENAS DE CARACTERES.**

Matrices unidimensionales.  
Matrices multidimensionales.  
Otras características de las matrices.  
Cadenas de caracteres. La clase string.  
Operaciones y métodos de la clase string.

### **TEMA 3.- HERENCIA.**

Herencia. Concepto y fundamentos.  
Modos de derivación.  
Jerarquía de clases.  
Vinculación entre métodos.  
Polimorfismo.  
Clases abstractas.  
Herencia múltiple en C++. Clases virtuales.

### **TEMA 4.- OTRAS CARACTERÍSTICAS DE C++.**

Sobrecarga de insertores y extractores.  
Funciones y clases amigas.  
Plantillas de funciones.  
Plantillas de clases.  
Manejo de errores.  
La biblioteca estándar de plantillas (STL):



#### BIBLIOGRAFÍA:

1. *Stroustrup, B.*  
**El lenguaje de programación C++.**  
Addison-Wesley 1998.
2. *Aho A. V., Hopcroft J. E., Ullman J. D.*  
**Estructuras de datos y algoritmos.**  
Addison-Wesley Iberoamericana 1988.
3. *Joyanes Aguilar L.*  
**Algoritmos y estructuras de datos (2ª edición).**  
McGraw Hill, 1996
4. *Heileman, G.L.,*  
**Estructuras de datos, algoritmos y programación orientada a objetos**  
McGraw Hill, 1998.
5. *Lafore, R.*  
**Object-Oriented Programming in C++.**  
SAMS Publishing 1999.
6. *Deitel, H.M., Deitel, P.J.*  
**Cómo programar en C++, (2ª edición).**  
Prentice Hall, 1999.
7. *Budd, T.A.*  
**An Introduction to Object-Oriented Programming, 2nd edition,**  
Addison-Wesley 1997. (Existe versión en castellano de la 1ª edición.)
8. *Budd, T.A.*  
**Classic Data Structures in C++.**  
Addison-Wesley 1994.