



1.Introducción.	2.Objetivos generales	3.Conocimientos previos requeridos	4.Contenidos mínimos.
5.Competencias.	6.Calendario de actividades.	7.Estimación de la dedicación del alumno.	8.Metodología.
9. Evaluación del aprendizaje.	10. Bibliografía		

## **1. Introducción.**

El objetivo de esta asignatura se centra en el conocimiento de las herramientas informáticas básicas para la organización y manipulación de datos. Para ello se presentan las estructuras de datos fundamentales y los diversos algoritmos clásicos que actúan sobre ellas. Mediante su estudio y análisis, además de conocer los métodos más usuales y sus características, se adquiere la técnica necesaria para la comprensión y evaluación de algoritmos más complejos y la capacidad de razonar cuál es la solución más adecuada para unas especificaciones concretas.

## **2. Objetivos Generales**

- Conocer las estructuras de datos fundamentales y los algoritmos principales que se utilizan para su manipulación.
- Conocer un lenguaje de programación de alto nivel y aplicarlo a la codificación de pequeños programas.
- Introducir el paradigma de la programación orientada a objetos.
- Estudiar y realizar las implementaciones de los TAD (Tipos Abstractos de Datos) a partir de su especificación.
- Comprender y analizar el concepto de eficiencia o complejidad en algoritmos básicos.
- Tener la capacidad de elección de la estructura de datos adecuada para cada tipo de problema.



### **3. Conocimientos previos.**

- a. Ser capaz de describir algoritmos en forma de pseudocódigo, independiente de un lenguaje final concreto, pero con precisión.
- b. Ser capaz de codificar de una manera correcta mediante el uso de estructuras de control claras, bucles, sentencias condicionales, etc. según convenga a la claridad y finalidad del segmento de código.
- c. Ser capaz de confeccionar, en un lenguaje estructurado, algoritmos correctos que resuelvan un problema de pequeña envergadura, expuesto, en términos de una especificación más o menos formal, y de decidir cuál de las posibles soluciones es la más apropiada para un entorno determinado.
- d. Saber agrupar conjuntos de instrucciones algorítmicas coherentemente mediante funciones o procedimientos.
- e. Conocer el mecanismo de paso de parámetros y utilizarlo correctamente.
- f. Conocer las estructuras de datos estáticas básicas.



#### **4. Contenidos mínimos.**

##### Tema 1. Repaso de Arrays y Cadenas.

- Inicialización.
  - Arrays de caracteres y de cadenas.
  - Arrays Multidimensionales.
  - Utilización de arrays como parámetros.
  - Lectura de cadenas.
  - Cadenas como parámetros y arrays de cadenas.
  - Procesamiento de cadenas (obtención de caracteres, comparación, conversión,..).
- Ejercicios: codificación en java de algoritmos que usen dichas estructuras.

##### Tema 2. Ordenación y Búsqueda

Se presentan y analizan (complejidad) los siguientes algoritmos de ordenación:

- Burbuja.
- Selección.
- Inserción.
- Shell.
- Ordenación rápida.
- Búsqueda secuencial y binaria.

Ejercicios: codificación en Java de dichos algoritmos. Comparativa de eficiencia.

##### Tema 3. Listas enlazadas.

- Fundamentos teóricos y clasificación.
- Operaciones con listas (inserción, búsqueda y borrado de nodos).
- Listas doblemente enlazadas y listas circulares.

Ejercicios. Codificación en Java de las operaciones sobre listas enlazadas.



#### Tema 4. Pilas y Colas.

- Concepto de Pila y especificación.
- Implementación de las operaciones básicas con pilas sobre arrays.
- Concepto de Cola y especificación.
- Implementación de operaciones básicas con colas sobre arrays.
- Implementación de pilas con listas enlazadas.
- Implementación de colas con listas enlazadas.

Ejercicios. Codificación en Java de los algoritmos básicos de trabajo con Pilas y Colas mediante listas enlazadas y arrays.

#### Tema 5. Árboles.

- Árboles generales.
- Árboles binarios, estructura y recorrido.
- Operaciones con árboles binarios.
- Árboles binarios de búsqueda. Operaciones (búsqueda, inserción, eliminación).

Ejercicios. Codificación en Java de las operaciones con árboles binarios de búsqueda.

#### Tema 6. Archivos.

- Archivos secuenciales y aleatorios.
- Algoritmos de ordenación de archivos: mergeSort.

Ejercicios. Codificación en Java de un algoritmo de ordenación de archivos.



### **5. Competencias genéricas.**

- a. Entender el funcionamiento y la utilidad de la gestión dinámica de memoria.
- b. Conocer el funcionamiento de la recursividad y ser capaz de construir algoritmos recursivos básicos.
- c. Desarrollar una aproximación disciplinada a la especificación, implementación, verificación y documentación de programas.
- d. Aprender el papel central que representa la abstracción en la tarea de programar.

### **6. Calendario de actividades**

Semana	Tema	Teoría(h)	Ejercicios(h)	Entrega Ej.	Sem(h)	Nota	Estudio	Totales
1 (16/02)	1	2					3	5
2	1	2	3				4	9
3	2	3		Tema 1	2	N1	4	9
4	2	2	3				4	9
5	2	2	3	Tema 2		N2	5	10
6	3	2			3		5	10
7	3	3	2				4	9
8	3		3	Tema 3	2	N3	5	10
9	4	3	2				5	10
10	4	3	2				5	10
11	5	3		Tema4	2	N4	5	10
12	5	2	3				5	10
13	5		3		2		5	10
14	6	3	2	Tema5		N5	5	10
15	6	2	3				5	10
16	6				3		4	7
Ex. Junio							5	5
Totales		32	29		14		73	148



## **7. Estimación de la dedicación del alumno.**

La penúltima columna de la tabla recoge el tiempo estimado de estudio y trabajo personal del alumno. Los seminarios se realizan al final de cada tema y no tienen por qué ser presenciales, se pueden realizar a través de alguna de las herramientas de Groupware o e-learning que se decida utilizar.

La dedicación del alumno podemos resumirla en lo siguiente:

- Clases teóricas, prácticas y seminarios: 69 h.
- Seminarios complementarios: 15 h.
- Examen Final: 5 h.
- Total de horas presenciales del alumno: 74 h.

Estimación de la dedicación no presencial de un alumno de tipo medio:

- Intervención en los seminarios: 15 h.
- Estudio y trabajo personal: 82 h.
- Total de horas con dedicación no presencial: 97 h.
- Dedicación total estimada del alumno: 148 h.

## **8. Metodología**

En el desarrollo de la asignatura se van a utilizar las siguientes técnicas:

- Exposición tradicional, por parte del profesor, de contenidos tales como la introducción de los temas, la descripción del funcionamiento de algoritmos típicos, y las soluciones a diversos problemas representativos. Se usará para ello la pizarra digital.
- Estudio y desarrollo o implementación de ejercicios y problemas propuestos por el profesor, que permitan ilustrar los contenidos mínimos y a la vez promuevan el ir más allá del conocimiento dado. Serán realizados por los alumnos en el laboratorio de informática.
- Puesta en común y análisis de las diferentes soluciones propuestas para los diferentes ejercicios, y de las posibles dificultades encontradas en su planteamiento y desarrollo.



Con el fin de facilitar la aplicación de esta metodología se va a recurrir al uso de una plataforma de e-learning tal como BSCW o Moodle, ya que aportan ventajas a tener en cuenta en el proceso de enseñanza-aprendizaje:

- Permiten crear entornos virtuales de trabajo, en los que los alumnos y el profesor colaboran y comparten los diversos recursos necesarios para el desarrollo del proceso formativo: documentación digital, páginas Web, artículos, ejercicios, soluciones, debates, agendas compartidas...
- Añaden utilidades que permiten realizar un seguimiento del trabajo del alumno.
- Aumenta las posibilidades de ajuste de la ayuda por parte del profesor al proceso de trabajo y aprendizaje de los estudiantes.
- Favorecer en los alumnos la capacidad de regular de manera más autónoma su proceso de aprendizaje
- Permiten trabajar e intercambiar información al margen de restricciones espacio-temporales.
- La documentación está permanentemente actualizada y accesible.
- Toda la interacción que se produce con la herramienta queda almacenada, de forma que facilita la evaluación de los alumnos.

El uso de una herramienta de este tipo, fomenta la interacción de los alumnos entre sí y con el profesor, al disponer de utilidades para la creación de debates en torno a temas o problemas específicos. Las intervenciones de los alumnos en los debates en vivo, cara a cara, no se producen con facilidad; sin embargo las discusiones mediadas se dan más fácilmente, y además, como las intervenciones quedan registradas, a menudo son más significativas al ser más reflexionadas.



## 9. Evaluación del aprendizaje.

Para evaluar el rendimiento académico de los alumnos se va a disponer de dos fuentes de datos:

- Las notas obtenidas a lo largo del cuatrimestre, y que tal y como refleja la hoja de actividades, son 5 (N1:N5) y corresponden a la evaluación de los ejercicios y actividades propuestas para cada uno de los temas. Dichas notas tendrán en cuenta las intervenciones de los alumnos en los correspondientes seminarios.
- La nota del examen final de Junio: NEJ

El cálculo de la nota final, NF, se efectúa en base al siguiente algoritmo:

Si  $NEJ \geq 5$

$$NF = \text{Suma}(N1:N5) * 0.6 + NEJ * 0.4$$

Si no

$$NF = NEJ$$

La nota de la convocatoria de Septiembre será la del examen correspondiente.





## 10. Bibliografía

- Documentación de la asignatura en:

<http://bscw.fit.fraunhofer.de/bscw/bscw.cgi/0/64310677>

- Aho, A.V., J.E. Hopcroft, J.D. Ullman, *Estructuras de datos y algoritmos*, Addison-Wesley.
- Joyanes, L., I. Zahonero, *Estructuras de Datos. Algoritmos, abstracción y objetos*, McGraw-Hill.
- Weiss, M.A., *Estructuras de datos y algoritmos*, Addison-Wesley.
- Wirth, N., *Algoritmos y Estructuras de Datos*, Prentice-Hall Iberoamericana, 1987.