

**Proyecto/Guía docente de la asignatura**

Asignatura	Fundamentos de Programación		
Materia	Proceso de desarrollo del SW		
Módulo	Formación Básica II		
Titulación	<ul style="list-style-type: none">Grado en Ingeniería Informática de Servicios y Aplicaciones (Grado IISA – Plan 413) ** Programa de estudios conjunto de Grado IISA y Grado en Matemáticas => <u>Doble Grado INFOMAT</u> : <ul style="list-style-type: none">Doble Grado INFOMAT (Itinerario desde Segovia – Plan 5472)		
Plan	413 / 5472	Código	40808
Periodo de impartición	Semestre 1	Tipo/Carácter	FB
Nivel/Ciclo	Grado	Curso	1
Créditos ECTS	6		
Lengua en que se imparte	Español		
Profesor/es responsable/s	Pilar Grande González		
Datos de contacto (E-mail, teléfono...)	Escuela de Ingeniería Informática Campus "María Zambrano" Plaza de la Universidad, 1 40.005 – Segovia (España) Teléfono: (+34) 921 11 24 51 Fax: (+34) 921 11 24 01 e-mail: pgrande@infor.uva.es		
Departamento	Informática (Área de Lenguajes y Sistemas Informáticos (LSI))		



1. Situación / Sentido de la Asignatura

1.1 Contextualización

La asignatura Fundamentos de la Programación pretende introducir a los alumnos en las técnicas de resolución de problemas de tipo general, mediante el empleo de los conceptos básicos y técnicas asociadas al paradigma de programación modular y estructurada de computadores, y la comprensión de la teoría subyacente al procesado de lenguajes de programación.

1.2 Relación con otras materias

Además de relacionarse con las otras asignaturas de la materia *Proceso de desarrollo del Software*, la asignatura FP también tiene relación directa con las materias *Matemáticas*, *Plataformas tecnológicas* e *Interacción Persona-Máquina*.

1.3 Prerrequisitos

No se han establecido. Debido a que la asignatura constituye el primer contacto con la programación, no se necesita ningún conocimiento previo sobre el tema. Por otro lado, no es prerrequisito de ninguna otra asignatura, pero es recomendable superar con éxito esta asignatura antes de cursar otras de la materia *Proceso de Desarrollo del Software*, como son la asignatura *Metodología de la Programación [MP]*, *Programación Orientada a Objetos [POO]* y la asignatura *Programación y Estructuras de Datos [PED]*. Superar la asignatura FP ayudará notablemente a comprender conceptos más complejos que serán presentados en las asignaturas citadas.

2. Competencias

2.1 Generales

- G01:** Conocimientos generales básicos.
- G02:** Conocimientos básicos de la profesión.
- G03:** Capacidad de análisis y síntesis.
- G04:** Capacidad de organizar y planificar.
- G05:** Comunicación oral y escrita en la propia lengua.
- G07:** Habilidades básicas en el manejo del ordenador.
- G08:** Habilidades de gestión de la información
- G09:** Resolución de problemas.
- G10:** Toma de decisiones.
- G11:** Capacidad crítica y autocrítica.
- G12:** Trabajo en equipo.
- G16:** Capacidad de aplicar los conocimientos en la práctica.
- G18:** Capacidad de aprender.
- G20:** Capacidad para generar nuevas ideas.
- G21:** Habilidad para trabajar de forma autónoma.
- G22:** Diseño y gestión de proyectos



2.2 Específicas

- E.3.** Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
- E.7.** Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.
- E.10.** Conocimiento, administración y mantenimiento sistemas, servicios y aplicaciones informáticas.
- E.11.** Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
- E.12.** Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
- E.13.** Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
- E.25.** Capacidad para comprender el entorno de una organización y sus necesidades en el ámbito de las tecnologías de la información y las comunicaciones.
- E.27.** Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas.
- E.31.** Capacidad para comprender, aplicar y gestionar la garantía y seguridad de los sistemas informáticos.
- E.34.** Capacidad para participar activamente en la especificación, diseño, implementación y mantenimiento de los sistemas de información y comunicación.

Estas competencias específicas se organizan de la siguiente forma:

Cognitivas (Saber)

1. Conocer y comprender la importancia de los objetivos de la programación.
2. Conocer los aspectos generales sobre los lenguajes y paradigmas de programación.
3. Comprender los fundamentos del paradigma de la programación modular y estructurada.
4. Conocer la sintaxis del pseudocódigo utilizado para describir algoritmos así como la del lenguaje de programación considerado en la asignatura.
5. Conocer los pasos para la realización de un programa y sus principales componentes.
6. Conocer las estructuras de control de la programación estructurada y las diferencias entre ellas (en pseudocódigo y en el lenguaje de programación considerado).
7. Conocer todos los aspectos relacionados con la realización de subprogramas (en pseudocódigo y en el lenguaje de programación considerado)
8. Conocer los tipos de datos estructurados estáticos (en pseudocódigo y en el lenguaje de programación considerado).
9. Aprender a utilizar un lenguaje de programación concreto y a transcribir a este lenguaje y ejecutar en una máquina real sus propios algoritmos.

Procedimentales/Instrumentales (Saber hacer):

1. Aprender técnicas básicas para la resolución de problemas mediante algoritmos.
2. A partir del planteamiento de un problema de pequeña-mediana envergadura saber realizar el programa para resolverlo implicando:
 - . Saber aplicar los pasos adecuados para la realización de programas.
 - . Tener en cuenta los objetivos de la programación.
 - . Saber elegir y utilizar los tipos y estructuras de datos adecuadas.
 - . Saber elegir y utilizar las estructuras de control adecuadas.
 - . Saber realizar la descomposición del problema de forma adecuada e implementar los subprogramas necesarios correctamente.
 - . Utilizar un estilo de programación apropiado.
3. Ser capaz de realizar el seguimiento de un algoritmo (en pseudocódigo) o programa (en el lenguaje de programación considerado), explicar qué tarea realiza y encontrar posibles errores. Ser capaz de proponer soluciones a los errores detectados.
4. Ser capaz de resolver pequeños algoritmos y programas de forma iterativa.



Actitudinales:

- Adquirir disciplina en la programación de computadores, siguiendo planteamientos que desarrollan la capacidad analítica del alumno para enfrentarse a problemas reales.
- Aprendizaje autónomo
- Planificación de las actividades a desarrollar
- Capacidad de abstracción
- Toma de decisión
- Capacidad de iniciativa y participación

3. Objetivos

- Comprender los conceptos fundamentales vinculados a la programación de computadores.
- Conocer y aplicar de forma adecuada los conceptos asociados al paradigma de programación modular y estructurada.
- Utilizar entornos de programación (IDE).

4. Contenidos y/o bloques temáticos

Bloque 1: Nociones fundamentales.

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

Este tema sirve para introducir las bases teóricas de la programación de computadores.

b. Objetivos de aprendizaje

- Comprender los conceptos fundamentales vinculados a la programación de computadores.

c. Contenidos

- Conceptos generales de la programación de computadores: algoritmos, programas, componentes de un ordenador, lenguajes de programación...
- Metodología para la resolución de problemas.

d. Métodos docentes

- 1.- Lección magistral: exposición de teoría
- 2.- Evaluación
- 3.- Estudio autónomo por parte del alumno, incluyendo consulta bibliográfica y preparación de pruebas de evaluación

e. Plan de trabajo

- Presentación en el aula de los conceptos teóricos asociados a este bloque temático.



f. Evaluación

- Estos conceptos formarán parte de la prueba escrita que se realizará al finalizar el bloque temático 4 así como en los exámenes teórico-prácticos finales ordinario y extraordinario de la asignatura. Ver tabla 7.

g. Bibliografía básica

- Kernighan, B.W., Ritchie, D.M. "El lenguaje de programación C". 2ª ed. (1991). México. Ed. Prentice-Hall Hispanoamericana.
- Schildt, H. "C: Manual de Referencia". (1997). 3ª ed. Ed. McGraw-Hill.
- Joyanes Aguilar, L., Zahonero Martínez, I. "Programación en C: metodología, algoritmos y estructuras de datos". 2ª ed. (2005). Madrid. Ed. McGraw-Hill.
- Gottfried, B.S.. "Programación en C". (2005). Madrid. Ed. McGraw-Hill.
- García Carballeira, F. "Problemas resueltos de programación en lenguaje C". (2002). Madrid. Ed. Thomson-Paraninfo.

h. Bibliografía complementaria

Pseudocódigo y diagramas de flujo:

- Joyanes Aguilar, L. "Fundamentos de Programación" 2ª ed. (1996). Ed. McGraw-Hill.
- Joyanes Aguilar, L., y otros. "Fundamentos de Programación. Libro de problemas"
- Carretero J., García F. y otros. "El lenguaje de programación C. Diseño e Interpretación de programas". (2002). Ed. Prentice-Hall.

i. Recursos necesarios

Aula con pizarra y ordenador con proyector, biblioteca, sala de estudio y despacho para tutorías.

j. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
0'5	



Bloque 2: Representación de Algoritmos.

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

En el tema anterior hemos presentado conceptos fundamentales para introducir al alumno en el proceso de programación de computadores. En este tema, avanzamos un poco más estudiando diferentes métodos de representación de algoritmos; concretamente, diagramas de flujo y pseudocódigo.

b. Objetivos de aprendizaje

- Ser capaz de representar un algoritmo mediante el correspondiente diagrama de flujo.
- Ser capaz de diseñar un algoritmo mediante pseudocódigo.

c. Contenidos

- Métodos de representación de algoritmos: Diagramas de flujo, pseudocódigo.

d. Métodos docentes

- 1.- Lección magistral: exposición de teoría
- 2.- Prácticas en aula: resolución de problemas
- 3.- Evaluación
- 4.- Estudio autónomo por parte del alumno, incluyendo la realización de problemas, consulta bibliográfica y preparación de pruebas de evaluación

e. Plan de trabajo

- Alternar sesiones teóricas con clases de problemas.

f. Evaluación

Ver tabla 7.

g. Bibliografía básica

- Kernighan, B.W., Ritchie, D.M. "El lenguaje de programación C". 2ª ed. (1991). México. Ed. Prentice-Hall Hispanoamericana.
- Schildt, H. "C: Manual de Referencia". (1997). 3ª ed. Ed. McGraw-Hill.
- Joyanes Aguilar, L., Zahonero Martínez, I. "Programación en C: metodología, algoritmos y estructuras de datos". 2ª ed. (2005). Madrid. Ed. McGraw-Hill.



- Gottfried, B.S.. "Programación en C". (2005). Madrid. Ed. McGraw-Hill.
- García Carballeira, F. "Problemas resueltos de programación en lenguaje C". (2002). Madrid. Ed. Thomson-Paraninfo.

h. Bibliografía complementaria

Pseudocódigo y diagramas de flujo:

- Joyanes Aguilar, L. "Fundamentos de Programación" 2ª ed. (1996). Ed. McGraw-Hill.
- Joyanes Aguilar, L., y otros. "Fundamentos de Programación. Libro de problemas"
- Carretero J., García F. y otros. "El lenguaje de programación C. Diseño e Interpretación de programas". (2002). Ed. Prentice-Hall.

i. Recursos necesarios

Aula con pizarra y ordenador con proyector, sala de ordenadores con el IDE considerado en la asignatura, biblioteca, sala de estudio y despacho para tutorías.

j. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
1	

Bloque 3: Componentes elementales de un lenguaje de programación.

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

En el tema anterior hemos aprendido a diseñar algoritmos. A partir de este tema, presentaremos la sintaxis y los principales elementos de un lenguaje de programación (en concreto, el lenguaje C) para comenzar a implementar pequeños programas que serán ejecutados en el computador.

b. Objetivos de aprendizaje

- Comprender los conceptos fundamentales vinculados a la programación de computadores.
- Conocer y aplicar de forma adecuada los conceptos asociados al paradigma de programación estructurada.
- Utilizar entornos de programación (IDE).



c. Contenidos

- Tipos de datos
- Operadores
- Expresiones
- Entrada y salida estándar

d. Métodos docentes

- 1.- Lección magistral: exposición de teoría
- 2.- Prácticas en aula: resolución de problemas
- 3.- Prácticas en el laboratorio: resolución de prácticas de laboratorio utilizando un IDE
- 4.- Evaluación
- 5.- Estudio autónomo por parte del alumno, incluyendo la realización de problemas, consulta bibliográfica, realización de prácticas y preparación de pruebas de evaluación

e. Plan de trabajo

- Alternar sesiones teóricas con prácticas y clases de problemas.
- Terminar con una práctica de ordenador que servirá de evaluación. Previamente, se realizará una sesión introductoria al software que se utilizará (IDE de lenguaje C).

f. Evaluación

Ver tabla 7.

g. Bibliografía básica

- Kernighan, B.W., Ritchie, D.M. "El lenguaje de programación C". 2ª ed. (1991). México. Ed. Prentice-Hall Hispanoamericana.
- Schildt, H. "C: Manual de Referencia". (1997). 3ª ed. Ed. McGraw-Hill.
- Joyanes Aguilar, L., Zahonero Martínez, I. "Programación en C: metodología, algoritmos y estructuras de datos". 2ª ed. (2005). Madrid. Ed. McGraw-Hill.
- Gottfried, B.S. "Programación en C". (2005). Madrid. Ed. McGraw-Hill.
- García Carballeira, F. "Problemas resueltos de programación en lenguaje C". (2002). Madrid. Ed. Thomson-Paraninfo.

h. Bibliografía complementaria

Pseudocódigo y diagramas de flujo:

- Joyanes Aguilar, L. "Fundamentos de Programación" 2ª ed. (1996). Ed. McGraw-Hill.
- Joyanes Aguilar, L., y otros. "Fundamentos de Programación. Libro de problemas"



- Carretero J., García F. y otros. "El lenguaje de programación C. Diseño e Interpretación de programas". (2002). Ed. Prentice-Hall.

i. Recursos necesarios

Aula con pizarra y ordenador con proyector, sala de ordenadores con el IDE considerado en la asignatura, biblioteca, sala de estudio y despacho para tutorías.

j. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
1	

Bloque 4: Estructuras de control.

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

Ampliamos conocimientos básicos de programación, presentando ahora las estructuras de control.

b. Objetivos de aprendizaje

- Comprender los conceptos fundamentales vinculados a la programación de computadores.
- Conocer y aplicar de forma adecuada los conceptos asociados al paradigma de programación estructurada.
- Utilizar entornos de programación (IDE).

c. Contenidos

- Estructura secuencial
- Estructura selectiva
- Estructura iterativa

d. Métodos docentes

- 1.- Lección magistral: exposición de teoría
- 2.- Prácticas en aula: resolución de problemas
- 3.- Prácticas en el laboratorio: resolución de prácticas de laboratorio utilizando un IDE
- 4.- Evaluación
- 5.- Estudio autónomo por parte del alumno, incluyendo la realización de problemas, consulta bibliográfica, realización de prácticas y preparación de pruebas de evaluación



e. Plan de trabajo

- Alternar sesiones teóricas con prácticas y clases de problemas.
- Terminar con una práctica de ordenador que servirá de evaluación.

f. Evaluación

Ver tabla 7.

g. Bibliografía básica

- Kernighan, B.W., Ritchie, D.M. "El lenguaje de programación C". 2ª ed. (1991). México. Ed. Prentice-Hall Hispanoamericana.
- Schildt, H. "C: Manual de Referencia". (1997). 3ª ed. Ed. McGraw-Hill.
- Joyanes Aguilar, L., Zahonero Martínez, I. "Programación en C: metodología, algoritmos y estructuras de datos". 2ª ed. (2005). Madrid. Ed. McGraw-Hill.
- Gottfried, B.S. "Programación en C". (2005). Madrid. Ed. McGraw-Hill.
- García Carballeira, F. "Problemas resueltos de programación en lenguaje C". (2002). Madrid. Ed. Thomson-Paraninfo.

h. Bibliografía complementaria

Pseudocódigo y diagramas de flujo:

- Joyanes Aguilar, L. "Fundamentos de Programación" 2ª ed. (1996). Ed. McGraw-Hill.
- Joyanes Aguilar, L., y otros. "Fundamentos de Programación. Libro de problemas"
- Carretero J., García F. y otros. "El lenguaje de programación C. Diseño e Interpretación de programas". (2002). Ed. Prentice-Hall.

i. Recursos necesarios

Aula con pizarra y ordenador con proyector, sala de ordenadores con el IDE considerado en la asignatura, biblioteca, sala de estudio y despacho para tutorías.

j. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
1	



Bloque 5: Subprogramación.

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

Ampliamos conocimientos básicos de programación, presentando ahora el concepto de subprograma.

b. Objetivos de aprendizaje

- Comprender los conceptos fundamentales vinculados a la programación de computadores.
- Conocer y aplicar de forma adecuada los conceptos asociados al paradigma de programación estructurada.
- Utilizar entornos de programación (IDE).

c. Contenidos

- Declaración y definición de funciones y procedimientos
- Paso de parámetros
- Librerías de funciones estándar

d. Métodos docentes

- 1.- Lección magistral: exposición de teoría
- 2.- Prácticas en aula: resolución de problemas
- 3.- Prácticas en el laboratorio: resolución de prácticas de laboratorio utilizando un IDE
- 4.- Evaluación
- 5.- Estudio autónomo por parte del alumno, incluyendo la realización de problemas, consulta bibliográfica, realización de prácticas y preparación de pruebas de evaluación

e. Plan de trabajo

- Alternar sesiones teóricas con prácticas y clases de problemas.
- Terminar con una práctica de ordenador que servirá de evaluación.

f. Evaluación

Ver tabla 7.



g. Bibliografía básica

- Kernighan, B.W., Ritchie, D.M. "El lenguaje de programación C". 2ª ed. (1991). México. Ed. Prentice-Hall Hispanoamericana.
- Schildt, H. "C: Manual de Referencia". (1997). 3ª ed. Ed. McGraw-Hill.
- Joyanes Aguilar, L., Zahonero Martínez, I. "Programación en C: metodología, algoritmos y estructuras de datos". 2ª ed. (2005). Madrid. Ed. McGraw-Hill.
- Gottfried, B.S. "Programación en C". (2005). Madrid. Ed. McGraw-Hill.
- García Carballeira, F. "Problemas resueltos de programación en lenguaje C". (2002). Madrid. Ed. Thomson-Paraninfo.

h. Bibliografía complementaria

Pseudocódigo y diagramas de flujo:

- Joyanes Aguilar, L. "Fundamentos de Programación" 2ª ed. (1996). Ed. McGraw-Hill.
- Joyanes Aguilar, L., y otros. "Fundamentos de Programación. Libro de problemas"
- Carretero J., García F. y otros. "El lenguaje de programación C. Diseño e Interpretación de programas". (2002). Ed. Prentice-Hall.

i. Recursos necesarios

Aula con pizarra y ordenador con proyector, sala de ordenadores con el IDE considerado en la asignatura, biblioteca, sala de estudio y despacho para tutorías.

j. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
1	

Bloque 6: Estructuras de datos.

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

Ampliamos conocimientos básicos de programación, presentando ahora unos tipos de datos más complejos que los vistos hasta el momento: las estructuras de datos.



b. Objetivos de aprendizaje

- Comprender los conceptos fundamentales vinculados a la programación de computadores.
- Conocer y aplicar de forma adecuada los conceptos asociados al paradigma de programación estructurada.
- Utilizar entornos de programación (IDE).

c. Contenidos

- Vectores
- Cadenas de caracteres
- Registros

d. Métodos docentes

- 1.- Lección magistral: exposición de teoría
- 2.- Prácticas en aula: resolución de problemas
- 3.- Evaluación
- 4.- Estudio autónomo por parte del alumno, incluyendo la realización de problemas, consulta bibliográfica, realización de prácticas y preparación de pruebas de evaluación

e. Plan de trabajo

- Alternar sesiones teóricas con prácticas y clases de problemas.
- Terminar con una práctica de ordenador que servirá de evaluación.

f. Evaluación

Ver tabla 7.

g. Bibliografía básica

- Kernighan, B.W., Ritchie, D.M. "El lenguaje de programación C". 2ª ed. (1991). México. Ed. Prentice-Hall Hispanoamericana.
- Schildt, H. "C: Manual de Referencia". (1997). 3ª ed. Ed. McGraw-Hill.
- Joyanes Aguilar, L. , Zahonero Martínez, I. "Programación en C: metodología, algoritmos y estructuras de datos". 2ª ed. (2005). Madrid. Ed. McGraw-Hill.
- Gottfried, B.S. "Programación en C". (2005). Madrid. Ed. McGraw-Hill.
- García Carballeira, F. "Problemas resueltos de programación en lenguaje C". (2002). Madrid. Ed. Thomson-Paraninfo.



h. Bibliografía complementaria

Pseudocódigo y diagramas de flujo:

- Joyanes Aguilar, L. "Fundamentos de Programación" 2ª ed. (1996). Ed. McGraw-Hill.
- Joyanes Aguilar, L., y otros. "Fundamentos de Programación. Libro de problemas"
- Carretero J., García F. y otros. "El lenguaje de programación C. Diseño e Interpretación de programas". (2002). Ed. Prentice-Hall.

i. Recursos necesarios

Aula con pizarra y ordenador con proyector, sala de ordenadores con el IDE considerado en la asignatura, biblioteca, sala de estudio y despacho para tutorías.

j. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
1'5	

5. Métodos docentes y principios metodológicos

- 1.- Lección magistral: exposición de teoría
- 2.- Prácticas en aula: resolución de problemas
- 3.- Prácticas en el laboratorio: resolución de prácticas de laboratorio utilizando un IDE
- 4.- Evaluación
- 5.- Estudio autónomo por parte del alumno, incluyendo la realización de problemas, consulta bibliográfica, realización de prácticas y preparación de pruebas de evaluación

6. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	30	Estudio y trabajo autónomo individual	60
Clases prácticas de aula (A)	--	Estudio y trabajo autónomo grupal	30
Laboratorios (L)	30		
Prácticas externas, clínicas o de campo	--		
Seminarios (S)	---		
Tutorías grupales (TG)	---		
Evaluación	---		
Total presencial	60	Total no presencial	90



7. Sistema y características de la evaluación

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Realización y defensa de varias prácticas de ordenador.	20%	Al finalizar los bloques temáticos 3, 4, 5 y 6. Estas prácticas deberán ser defendidas ante el profesor. La nota final obtenida en esta parte debe ser ≥ 5 (sobre un total de 10 puntos), para que se considere aprobada la <u>parte práctica</u> de la asignatura.
Realización de exámenes escritos de carácter teórico-práctico.	80%	La nota final obtenida en esta parte debe ser ≥ 5 (sobre un total de 10 puntos), para que se considere aprobada la <u>parte teórica</u> de la asignatura.
Importante: Para aprobar la asignatura es necesario aprobar ambas partes (teórica y práctica) por separado.		

CRITERIOS DE CALIFICACIÓN

- **Convocatoria ordinaria:**
 - Ver tabla anterior
- **Convocatoria extraordinaria:**
 - Ver tabla anterior

8. Consideraciones finales

Durante el desarrollo de la asignatura y vinculadas directamente con los conceptos teóricos impartidos en el aula, se realizan varias prácticas en el laboratorio lo que supone un trabajo continuo para el alumno.

La evaluación final de la asignatura se llevará a cabo en las convocatorias finales (ordinaria y extraordinaria), y consistirá en la realización de un examen teórico-práctico sobre los 6 bloques temáticos de la asignatura, siendo preciso además, realizar a lo largo del cuatrimestre la entrega y defensa de las prácticas de laboratorio propuestas por el profesor, en las fechas que se establezcan y que serán comunicadas con la suficiente antelación a los alumnos a través del campus virtual (manteniéndose la misma ponderación detallada con anterioridad: 80% para el examen teórico-práctico y 20% para la parte de prácticas).