Proyecto/Guía docente de la asignatura

Asignatura	Metodología de la Programación		
Materia	Proceso de desarrollo del SW		
Módulo	Formación Básica II		
Titulación	** Programa de estudios conjunto de Grado IISA y Grado en Matemáticas => • Doble Grado INFOMAT (Itinerario desde Segovia)		
Plan	5472	**Código**	40847
Periodo de impartición	Semestre 2	**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) Despacho: D237 (Fase II del campus) 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 Metodología de la Programación *[MP]* complementa los conceptos introducidos en la asignatura *"Fundamentos de Programación" [FP]* que se imparte en el semestre 1. En esta ocasión, los alumnos profundizan en el estudio de las técnicas de resolución de problemas de tipo general, mediante el empleo de los conceptos 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. Se tratan conceptos más avanzados relacionados con la gestión de archivos, el análisis de la eficiencia de los algoritmos, la recursividad y la asignación dinámica de memoria, entre otros.

1.2 Relación con otras materias

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

1.3 Prerrequisitos

No es prerrequisito de ninguna otra asignatura, pero es recomendable haber superado con éxito la asignatura "*Fundamentos de Programación" [FP]* para comprender conceptos más complejos que serán presentados en ésta. Además, el alumno debe tener en cuenta que se recomienda superar con éxito esta asignatura antes de cursar otras de la materia *Proceso de Desarrollo del Software*, como son la asignatura "*Programación Orientada a Objetos" [POO]*, y la asignatura "*Programación y Estructuras de Datos" [PED].* Superar la asignatura MP 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 el concepto y las características fundamentales del tipo de datos persistente (fichero) y cómo se trabaja con él en el mundo de la programación de computadores.
5. Conocer el concepto de recursividad y sus tipos.
6. Comprender el concepto de memoria dinámica y aprender a trabajar en ella con ayuda de punteros.
7. 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. Saber analizar la eficiencia de los algoritmos.
5. Ser capaz de resolver pequeños algoritmos y programas tanto iterativa como recursivamente.

Actitudinales:

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

3. Objetivos

- Comprender conceptos básicos y avanzados vinculados a la programación de computadores.
- Conocer y aplicar de forma adecuada los conceptos asociados al paradigma de programación modular y estructurada.
- Conocer técnicas básicas de prueba de programas.
- Utilizar entornos de programación (IDE).

4. Contenidos y/o bloques temáticos

Bloque 1:Unidad I: Gestión de archivos

Tema 1.- Ficheros

Carga de trabajo en créditos ECTS: 1

a. Contextualización y justificación

Hasta el momento, toda la información que hemos gestionado en nuestros programas o eran datos del propio programa, o eran datos que introducía el usuario desde el teclado. Por otro lado, siempre que un programa ha obtenido un resultado nos hemos limitado a mostrarlo en pantalla. En esta unidad introducimos el concepto de persistencia asociada a los datos. Veremos cómo es posible almacenar de forma permanente la información generada por un programa de modo que esta información se encuentre a nuestra disposición entre sucesivas ejecuciones del mismo. Por tanto, en esta ocasión, ampliamos conocimientos básicos de programación, presentando un tipo de datos con persistencia, el archivo o fichero, cuyo tiempo de vida no está ligado al de la ejecución del programa que lo crea o lo maneja.

b. Objetivos de aprendizaje

- Comprender los conceptos fundamentales vinculados a la programación de computadores.
- Comprender el concepto de fichero y aprender cómo se realiza el procesamiento de los mismos.
- Conocer los conceptos básicos relacionados con el tipo de datos estructurado denominado fichero (o archivo).
- Desarrollar aplicaciones que trabajen en conjunto con estructuras de datos de distintos tipos: ficheros, arrays, cadenas de caracteres, registros..
- Utilizar entornos de programación (IDE).

c. Contenidos

- Tipo de dato archivo o fichero.
- Tipos de ficheros: estructura que presentan.
- Modos de acceso a un fichero.
- Procesamiento de ficheros.

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 apdo. 7

g Material docente

g.1 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: metodologia, 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.

g.2 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.

g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), …)

---

h. 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.

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
1	

Bloque 2:Unidad II: Análisis de la eficiencia de los algoritmos.

Tema 2.- Técnicas de análisis de algoritmos.

Carga de trabajo en créditos ECTS: 1

a. Contextualización y justificación

Ampliamos conocimientos básicos de programación adquiridos en la asignatura *"Fundamentos de programación" [FP]* presentando los conceptos fundamentales asociados al análisis de la eficiencia de los algoritmos, de forma que los alumnos dispongan de recursos que le permitan decidir qué algoritmo resulta más adecuado entre distintos disponibles para resolver un determinado problema.

b. Objetivos de aprendizaje

- Realizar el análisis de la eficiencia de algoritmos.

c. Contenidos

Estudio de los conceptos básicos asociados al análisis de la eficiencia de los algoritmos.

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

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

f. Evaluación

- Ver tabla apdo. 7

g Material docente

g.1 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: metodologia, 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.

g.2 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.

g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), …)

---

h. Recursos necesarios

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

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
1	

Bloque 3:Unidad III: Recursividad

Tema 3.- Introducción a la recursividad.
Tema 4.- Algoritmos recursivos de ordenación y búsqueda en vectores.

Carga de trabajo en créditos ECTS: 1'5

a. Contextualización y justificación

Se presenta una técnica de diseño de algoritmos fundamental en el mundo de la programación de computadoras: la recursividad. Estudiaremos cómo se diseña un algoritmo recursivo y sentaremos las bases teóricas que nos permitirán definir estructuras de datos recursivas en la asignatura *"Programación y Estructuras de Datos" [PED].*

b. Objetivos de aprendizaje

- Comprender el concepto de recursividad.
- Identificar distintos tipos de recursividad.
- Ser capaz de realizar el diseño e implementación de un algoritmo recursivo.

c. Contenidos

- Estudio detallado de la recursividad como técnica de diseño de programas. Puesta en práctica de estos conceptos con ayuda de los principales algoritmos recursivos de búsqueda y ordenación de vectores.

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 clases de problemas.
- Realización de una práctica de laboratorio que servirá de evaluación.

f. Evaluación

- Ver tabla apdo. 7

g Material docente

g.1 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: metodologia, 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.

g.2 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.

g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), …)

---

h. 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.

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
1’5	

Bloque 4:Unidad IV: Memoria dinámica

Tema 5.- Punteros.
Tema 6.- Asignación dinámica de memoria.

Carga de trabajo en créditos ECTS: 2

a. Contextualización y justificación

Hasta ahora hemos trabajado con la memoria estática del computador. En este tema se plantea la reserva dinámica de memoria y la referencia a variables a través de su dirección de memoria. Las técnicas de programación moderna exigen tratar con estructuras de datos dinámicas que, al contrario de las estáticas, se almacenan en áreas de la memoria de tamaño variable ya que su número crece o decrece en función de las necesidades del problema concreto. La utilización de variables dinámicas se realiza a través de un tipo especial de variable denominado apuntador o puntero, que permite representar direcciones de memoria de un dato de un tipo determinado. Aprenderemos a realizar reserva de espacio en memoria dinámica, trabajaremos en ese espacio reservado y finalmente, procederemos a realizar la liberación del mismo.

b. Objetivos de aprendizaje

- Comprender los conceptos fundamentales vinculados a la programación de computadores.
- Conocer los conceptos básicos relacionados con el tipo de datos puntero.
- Utilizar entornos de programación (IDE).

c. Contenidos

- Tipo de datos puntero
- Concepto de variable dinámica
- Trabajo con punteros

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.
- Opcional: Terminar con una práctica de ordenador que servirá de evaluación.

f. Evaluación

- Ver tabla apdo. 7

g Material docente

g.1 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: metodologia, 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.

g.2 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.

g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), …)

---

h. 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.

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
2	

Bloque 5:Unidad V: Prueba de programas

Tema 7.- Técnicas de prueba de programas.

Carga de trabajo en créditos ECTS: 0'5

a. Contextualización y justificación

La prueba de programas es un proceso muy costoso. Para justificarlo debemos aumentar la fiabilidad del software probado. Sólo podemos garantizar el aumento de la fiabilidad si localizamos errores ocultos en el software. En esta unidad aprenderemos diferentes técnicas que nos van a permitir llevar a cabo la prueba de nuestros programas.

b. Objetivos de aprendizaje

- Comprender los conceptos fundamentales asociados a la prueba de programas.
- Conocer los principales tipos de pruebas.
- Utilizar entornos de programación (IDE).

c. Contenidos

- Conceptos de prueba de un programa.
- Tipos de Pruebas.

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.

f. Evaluación

- Ver tabla apdo. 7

g Material docente

g.1 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: metodologia, 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.

g.2 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.

g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), …)

---

h. 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.

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
0’5	

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 o PRESENCIALES A DISTANCIA(1)	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**
		TOTAL presencial + no presencial	**150**

(1) Actividad presencial a distancia es cuando un grupo sigue una videoconferencia de forma síncrona a la clase impartida por el profesor.

7. Sistema y características de la evaluación

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Realización y defensa de varias prácticas de ordenador.	25%	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 parte práctica de la asignatura.
Realización de exámenes escritos de carácter teórico-práctico.	75%	La nota final obtenida en esta parte debe ser >= 5 (sobre un total de 10 puntos), para que se considere aprobada la parte teórica 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 5 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: 75% para el examen teórico-práctico y 25% para la parte de prácticas).