

PLAN DOCENTE DE LA ASIGNATURA

Curso académico: 2025/2026

Identificación y características de la asignatura					
Código	502304, 503338				
Denominación (español)	Introducción a la Programación				
Denominación (inglés)	Introduction to Programming				
Titulaciones	Grado en Ingeniería Informática en Ingeniería de Computadores Grado en Ingeniería Informática en Ingeniería del Software Doble Grado en Administración y Dirección de Empresas e Ingeniería Informática en Ingeniería de Computadores Doble Grado en Administración y Dirección de Empresas e Ingeniería Informática en Ingeniería del Software				
Centro	Escuela Politécnica				
Módulo	Formación básica				
Materia	Informática				
Carácter	Formación básica	ECTS	6	Semestre	1
Profesorado					
Nombre		Despacho		Correo-e	
Alberto Gómez Mancha		10		agomez@unex.es	
M ^a Ángeles Mariscal Araujo		17		mariscal@unex.es	
Fernando Sánchez Figueroa		59		fernando@unex.es	
M ^a Encarnación Sosa Sánchez		41		esosa@unex.es	
Cristina Vicente Chicote		42		cristinav@unex.es	
Área de conocimiento	Lenguajes y Sistema Informáticos				
Departamento	Ingeniería de Sistemas Informáticos y Telemáticos				
Profesor coordinador	Alberto Gómez Mancha				
Competencias					
Competencias básicas (comunes a todas las enseñanzas de Grado)					
<i>Se recogen por defecto</i>					
CB1: Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.					
CB2: Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.					

CB3: Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

CB4: Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CB5: Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

Competencias generales

CG02: Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la Informática de acuerdo con los conocimientos adquiridos para la tecnología específica de Ingeniería de Computadores o de Ingeniería del Software.

CG04 - Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas, de acuerdo con los conocimientos adquiridos para la tecnología específica de Ingeniería de Computadores o de Ingeniería del Software.

CG05 - Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos para la tecnología específica de Ingeniería de Computadores o de Ingeniería del Software.

CG08 - Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

CG09 - Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero/a Técnico/a en Informática.

CG10 - Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática, de acuerdo con los conocimientos adquiridos para la tecnología específica de Ingeniería de Computadores o de Ingeniería del Software.

Competencias específicas del módulo de formación básica

Según los planes de estudio aprobados, esta asignatura debe cubrir, parcialmente, las siguientes competencias técnicas y sus resultados de aprendizaje.

CFB03: Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.

CFB04: Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

Competencias transversales

Según los planes de estudio aprobados y los acuerdos de la comisión de calidad de las titulaciones, esta asignatura debe cubrir, total o parcialmente, las siguientes competencias transversales y sus resultados de aprendizaje en un nivel básico.

CT09: Capacidad de trabajo en grupo.

CT14: Orientación a la calidad y a la mejora continua.

Contenidos

Descripción general del contenido: Fundamentos de programación y de estructuras de datos básicas, aplicaciones y propiedades. Características principales del paradigma imperativo.

Diseño, implementación y uso de tipos abstractos de datos. Lenguajes estructurados para el desarrollo de sistemas software.

Temario

Tema 1: Introducción

1. Conceptos básicos: algoritmo, proceso y procesador
2. Estructura y funcionamiento básicos de un ordenador
3. Codificación y almacenamiento de la información
4. Representación de algoritmos

Tema 2: Primeros algoritmos

1. Entorno: elementos básicos
2. Variables. Tipos simples de datos
3. Operaciones y expresiones
4. Acciones elementales
5. Metodología de resolución de problemas
6. Representación en C++

Tema 3: Programación modular

1. Diseño descendente
2. Módulos: concepto, tipos, parámetros
3. Especificación con pre- y post- condiciones
4. Representación en C++

Tema 4: Programación estructurada

1. Estructuras básicas de control
2. Otras estructuras de control
3. Representación en C++

Tema 5: Recursividad

1. Conceptos básicos. Clasificación
2. Problemas

Tema 6: Análisis de algoritmos

1. Análisis de la eficiencia. Tiempo y espacio. Caso mejor, peor y medio
2. Órdenes de complejidad. Notación O

Tema 7: Estructuras de datos y TAD

1. Tipos abstractos de datos. Especificación
2. Estructuras de datos
3. Representación en C++

Tema 8: Registros

1. Definición
2. Operaciones básicas
3. Representación en C++

Tema 9: Vectores

1. Definición y operaciones básicas
2. Algoritmos de búsqueda y de ordenación
3. Cadenas de caracteres

4. TAD con implementación lineal y estática
5. Especificación e implementaciones del TAD Pila, Cola, Lista

Tema 10: Gestión dinámica de memoria

1. Punteros y variables dinámicas
2. Representación en C++
3. Listas enlazadas
4. Cadenas de caracteres
5. TAD con implementación lineal y dinámica
6. Implementaciones del TAD Pila, Cola y Lista

En las sesiones de laboratorio se irán realizando ejercicios y resolviendo problemas aplicando los contenidos teóricos vistos previamente.

Actividades formativas

Horas de trabajo del alumno/a por tema		Horas Gran grupo	Actividades prácticas				Actividad de seguimiento	No presencial
Tema	Total	GG	CH	L	O	S	TP	EP
1	3	1		0				2
2	6.5	1		1.5				4
3	12.5	3		1.5				8
4	25	6		3				16
5	11	1.5		1.5				8
6	7.5	2		1.5				4
7	10	1		3				6
8	10	1		3				6
9	32	11		3				18
10	28	7		3				18
Evaluación	4.5	3		1.5				0
TOTAL	150	37.5	0	22.5	0	0	0	90

GG: Grupo Grande (85 estudiantes).

CH: Actividades de prácticas clínicas hospitalarias (7 estudiantes)

L: Actividades de laboratorio o prácticas de campo (15 estudiantes) **(En esta asignatura y en otras de esta titulación, por orden del Vicerrectorado de Profesorado, grupos de 20 estudiantes.)**

O: Actividades en sala de ordenadores o laboratorio de idiomas (20 estudiantes)

S: Actividades de seminario o de problemas en clase (40 estudiantes).

TP: Tutorías Programadas (seguimiento docente, tipo tutorías ECTS).

EP: Estudio personal, trabajos individuales o en grupo, y lectura de bibliografía.

Metodologías docentes

En el desarrollo de la asignatura se combinarán distintas actividades que permitan la participación.

A continuación, se nombran algunas de las actividades formativas que se plantearán a lo largo del curso para alcanzar los resultados de aprendizaje de la asignatura.

Aunque cada actividad sólo se detalla dentro de una modalidad (presenciales en grupo grande, presenciales en laboratorio y no presenciales), algunas de ellas se desarrollarán en varias.

Algunas de estas actividades se realizarán de forma individual y otras, en grupo.

Presenciales en grupo grande

- Clase expositiva
- Clase de explicación de ejercicios y problemas
- Método del caso

- Resolución de tests
- Resolución de ejercicios y problemas
- Comparación de algoritmos
- Trabajo en grupo para alcanzar acuerdos y desarrollar problemas en común
- Análisis de representación de tipos abstractos de datos
- Autoevaluación, con aplicación de rúbrica
- Evaluación entre pares, con aplicación de rúbrica
- Resolución de cuestionarios
- Ensayo de examen
- Exámenes de problemas
- Presentación de problemas resueltos

Presenciales en laboratorio

- Clase de demostración de software
- Trazas de algoritmos
- Laboratorios guiados
- Laboratorios abiertos
- Portafolio de actividades
- Prueba y detección de errores en programas
- Análisis de documentación interna y externa de programas
- Uso de librerías de programación
- Detección de requisitos de programas
- Diseño y justificación de estructuras de datos
- Proyecto de programación
- Modificación de programas para incorporar nuevas funcionalidades
- Subida de ficheros al aula virtual
- Revisión de portafolio

No presenciales

- Estudio individual
- Reuniones de grupo
- Búsqueda de información
- Plantear preguntas de test
- Elaboración de listas de conceptos
- Definición de conceptos
- Seguimiento de problemas resueltos
- Estudio inicial de temas
- Implementación de programas
- Escritura de documentación interna de programas
- Acceso a documentación del aula virtual
- Consulta de páginas web
- Visualización de videos de funcionamiento de herramientas, de explicación de conceptos y de resolución de problemas
- Construcción colaborativa de glosarios en el aula virtual
- Comunicación con profesorado y estudiantes mediante foros
- Cuestionarios de evaluación y autoevaluación del trabajo en grupo

Resultados de aprendizaje

Según la memoria de verificación del título, los resultados de aprendizaje asociados a las competencias que desarrolla la materia que incluye a esta asignatura son los siguientes:

- RAE1. Conoce los elementos básicos de la programación (paradigmas, estructuras de control, estructuras de datos, lenguajes, algoritmos, complejidad, recursividad, etc.) y los aplica de forma eficiente y correcta en la resolución de problemas.
- RAE2. Comprende los conceptos básicos de la programación imperativa y la programación orientada a objetos, diferencia ambos paradigmas y puede aplicarlos en un problema de manera conveniente.
- RAE3. Conoce y utiliza los tipos abstractos de datos fundamentales, y puede implementarlos en un lenguaje de alto nivel imperativo u orientado a objetos.
- RAE4. Diseña, implementa, prueba y documenta programas de pequeño tamaño atendiendo a criterios de eficacia y legibilidad.

Según los acuerdos de la comisión de calidad, los resultados de aprendizaje asociados a las competencias transversales que deben desarrollarse en la materia que incluye a esta asignatura son los siguientes:

- RAC1. Conoce las características básicas del trabajo en grupo y las técnicas elementales de trabajo en equipo.
- RAC2. Llega a acuerdos con los componentes de su grupo para alcanzar los objetivos comunes.
- RAC3. Colabora activamente en la organización y realización de las tareas del grupo.
- RAC4. Sigue las normas establecidas en las actividades y propone mejoras.
- RAC5. Planifica sus actuaciones con el objetivo de asegurar y mejorar la calidad de lo que hace.

A partir de esos resultados generales de la materia, se han establecido los siguientes resultados de aprendizaje concretos de esta asignatura:

- RA01. Explicar el funcionamiento de código escrito en un lenguaje de programación estructurado.
- RA02. Implementar algoritmos que manejen estructuras de datos básicas en memoria interna para resolver problemas simples.
- RA03. Seleccionar las estructuras de datos y los algoritmos básicos sobre ellas más adecuados para definir tipos abstractos de datos eficientes.
- RA04. Utilizar las técnicas y herramientas software básicas empleadas en el diseño y desarrollo de programas.
- RA05. Desarrollar programas de tamaño medio, escritos en un lenguaje estructurado, aplicando todas las etapas de un método de construcción de software, individualmente y en grupo.
- RA06. Comparar distintas versiones de algoritmos y programas para seleccionar el más adecuado al problema aplicando criterios de eficiencia y calidad.

Relación entre las competencias de la asignatura y los resultados de aprendizaje

	CFB03	CFB04	CT09	CT14
RA01	X	X		
RA02	X	X		
RA03	X	X	X	X
RA04		X		
RA05	X	X	X	X
RA06	X	X		X

Relación entre los resultados de aprendizaje del título y los resultados de aprendizaje de la asignatura

	RAE1	RAE2	RAE3	RAE4	RAC1	RAC2	RAC3	RAC4	RAC5
RA01	X	X		X					
RA02	X	X	X	X					
RA03	X	X	X	X	X	X	X	X	X
RA04			X	X					
RA05	X	X		X	X	X	X	X	X
RA06	X			X				X	X

Sistemas de evaluación

Tal como se contempla en la vigente normativa de evaluación de la Universidad de Extremadura, esta asignatura puede superarse siguiendo las modalidades de evaluación continua (E.C.) y de evaluación global (E.G.).

La elección de la modalidad de evaluación corresponde a cada estudiante. Se podrá elegir el sistema de evaluación de cada convocatoria en una consulta del aula virtual de la asignatura durante las cuatro primeras semanas del curso. Si no se comunica el tipo de evaluación elegido en el plazo indicado se supondrá que se opta por la modalidad de evaluación continua.

A continuación, se detallan las características de ambos sistemas de evaluación.

Instrumentos de evaluación

Para poder evaluar la consecución de los objetivos de aprendizaje de esta asignatura, tanto los relacionados con las competencias técnicas como los que corresponden a las competencias transversales, se han considerado adecuados los siguientes instrumentos de evaluación:

- Actividades
- Proyecto de programación
- Pruebas escritas

Estos son instrumentos generales que integran otros instrumentos de evaluación más directos y simples y que permiten realizar un análisis completo de los niveles de consecución de las distintas competencias.

Actividades

El portafolio de actividades está formado por distintas actividades simples realizadas por cada estudiante a lo largo del semestre, recopilando trabajos realizados en las clases presenciales (en grupo grande y en laboratorio) o durante su trabajo personal.

Estas actividades pueden ser de muchos tipos: resolución de problemas, resolución de tests, propuestas de nuevos problemas, corrección de trabajos de estudiantes, búsqueda de información, programas, actividades del aula virtual, trabajos en grupo, actas de reuniones, mapas conceptuales, etc.

Además del valor individual de cada actividad incluida en el portafolio como herramienta de evaluación, el portafolio tiene un valor adicional al permitir ver la evolución del trabajo realizado a lo largo del curso, reflexionar sobre la evolución y aplicar las medidas correctoras necesarias para mejorar y alcanzar el nivel deseado al final del semestre.

Todas las competencias que se evalúan en estas actividades se evalúan, además, en los dos instrumentos de evaluación siguientes. Estas actividades sirven, principalmente, para fomentar el trabajo continuo en la asignatura y para recibir *feedback* a lo largo del semestre.

Proyecto de programación

El proyecto de programación es un instrumento de evaluación que permite evaluar muchas de las competencias técnicas y transversales de la futura graduada o graduado en Ingeniería Informática, al tratarse de una actividad muy próxima a la que deberá enfrentarse regularmente durante su vida laboral: la resolución de un problema planteado mediante un sistema software y la documentación necesaria.

En este caso, el problema presentado se adecuará al nivel inicial de la asignatura, proporcionándose partes ya terminadas y dirigiéndose la construcción del software con las instrucciones necesarias.

Pruebas escritas

Para asegurar la adquisición de los conocimientos y habilidades mínimos de las competencias técnicas se realizarán pruebas escritas que consistirán en la resolución de problemas, preguntas de tipo test, preguntas cortas, etc.

Relación entre instrumentos de evaluación y resultados de aprendizaje

En la siguiente tabla se detallan los resultados de aprendizaje de la asignatura que se cubren con los instrumentos de evaluación propuestos.

<i>Resultados de aprendizaje</i>	Instrumentos de evaluación		
	Actividades (15% en E.C.; 0% en E.G.)	Proyecto (35%)	Pruebas (50% en E.C.; 65% en E.G.)
RA01	X		X
RA02	X	X	X
RA03	X	X	X
RA04	X	X	
RA05	X	X	
RA06	X		X

Criterios de evaluación

- Esta asignatura se puede superar según dos modalidades de evaluación distintas: evaluación continua (E.C.) o evaluación global (E.G.).

Modalidad de evaluación continua:

- Para superar esta asignatura por evaluación continua deben superarse los requisitos mínimos de los 3 bloques de la asignatura (Actividades, Proyecto y Pruebas), asociados a los tres instrumentos de evaluación principales (actividades, proyecto de programación y pruebas escritas, respectivamente).

- La puntuación de cada bloque se calculará sobre 10.

- La nota en un bloque superado (cumplidos los requisitos mínimos) se guardará durante todas las convocatorias del curso (siempre que se tenga derecho a examen en la convocatoria en la que se supere el bloque).

Bloque 1: Actividades

- La nota del bloque de actividades, NACT, representa el 15% de la nota final.

- Este bloque se superará realizando una serie de actividades que se irán proponiendo a lo largo del curso. Según su naturaleza, se realizarán en las clases de teoría, laboratorio o en horario no presencial usando el aula virtual.

- La asistencia a las clases de teoría y a las de laboratorio podrán considerarse, a efectos de evaluación, como actividades.

- La nota de este bloque se obtiene a partir de la ponderación de las calificaciones obtenidas en las actividades propuestas.
- No es necesario obtener una nota mínima en este bloque para considerarlo superado.
- La nota obtenida en este bloque no es recuperable en las convocatorias extraordinarias del curso.

Bloque 2: Proyecto

- La nota del bloque de proyecto, NPRO, representa el 35% de la nota.
- Es obligatorio superar este bloque con una nota mínima de 5 sobre 10.
- Es responsabilidad de cada estudiante la custodia y protección de su proyecto.
- Son requisitos indispensables para superar este bloque: entregar el programa y la documentación solicitados cumpliendo los requisitos mínimos que se establezcan, superar la defensa del proyecto y utilizar adecuadamente el sistema de desarrollo.
- La nota de este bloque es recuperable en las convocatorias extraordinarias de la asignatura, para lo que será necesario presentar el proyecto y superar la defensa del proyecto.

Bloque 3: Pruebas

- La nota del bloque de pruebas escritas, NPRU, representa el 50% de la nota.
- Para superar este bloque es necesario obtener una nota mínima de 5 sobre 10.
- Este bloque podrá superarse si se aprueban todas las pruebas parciales que se plantearán a lo largo del curso. En la convocatoria ordinaria se podrá recuperar de forma independiente cada uno de los parciales. En el resto de las convocatorias extraordinarias se realizará una prueba global de la materia para recuperar este bloque.
- Cada prueba, parcial o final, podrá estar compuesto de preguntas de test o de respuestas cortas o resolución de problemas, con requisitos adicionales sobre la nota mínima que debe obtenerse en cada parte para poder hacer media.

Modalidad de evaluación global:

- Para superar esta asignatura con evaluación global deben superarse los requisitos mínimos de las dos partes de las que constará el examen de la convocatoria: Proyecto y Pruebas.
- La puntuación de cada parte se calculará sobre 10.
- La nota de una parte superada no se guardará para ninguna convocatoria posterior del curso.

Parte 1: Proyecto

- La nota de la parte de proyecto, NPRO, representa el 35% de la nota.
- Es obligatorio superar esta parte con una nota mínima de 5 sobre 10.
- Son requisitos indispensables para superar esta parte: entregar el programa y la documentación solicitados cumpliendo los requisitos mínimos que se establezcan, superar la defensa del proyecto y utilizar adecuadamente el sistema de desarrollo.

Parte 2: Pruebas

- La nota de la parte de pruebas escritas, NPRU, representa el 65% de la nota.
- Para superar esta parte es necesario obtener una nota mínima de 5 sobre 10.
- Esta prueba podrá estar compuesta de preguntas de test o de respuestas cortas o resolución de problemas, con requisitos adicionales sobre la nota mínima que debe obtenerse en cada problema para poder hacer media.

El proyecto de programación, la defensa del proyecto y las pruebas escritas finales serán las mismas, independientemente del modelo de evaluación que se haya seleccionado.

Cálculo de la nota final de la asignatura

- La copia o el plagio en cualquier actividad, prueba o proyecto, ya sea en una parte o en su totalidad, supone una nota final de SUSPENSO (0) en la convocatoria y una nota de 0 en todas las calificaciones obtenidas hasta el momento para todas las personas implicadas, además de las actuaciones legales pertinentes.

- En el caso de la evaluación continua, si se cumplen todos los requisitos mínimos de los tres bloques, la nota final se calcula como la siguiente media ponderada:

$$\text{NotaFinal} = 0.15 * \text{NACT} + 0.35 * \text{NPRO} + 0.50 * \text{NPRU}$$

- En el caso de la evaluación global, si se cumplen todos los requisitos mínimos de las dos partes, la nota final se calcula como la siguiente media ponderada:

$$\text{NotaFinal} = 0.35 * \text{NPRO} + 0.65 * \text{NPRU}$$

- Si se sigue la modalidad de evaluación continua y se cumplen los requisitos mínimos de los bloques, la nota final se calculará con la fórmula de nota final con la que se obtenga un resultado más alto.

- Si no se cumplen todos los requisitos mínimos, la nota final será 2 o 3, dependiendo de los casos.

En la siguiente tabla se puede consultar cómo calcular la nota final, según los resultados obtenidos en los dos bloques obligatorios de proyecto y pruebas escritas:

Pruebas	NP	NP	NP	NCR	NCR	Nota	Nota
Proyecto	NP	NCR	Nota	NP	NCR/Nota	NP/NCR	Nota
Nota final *	NP	2	3	2	3	3	Nota

NP: no se ha presentado a ese bloque

NCR: no se cumplen los requisitos mínimos de ese bloque

Nota: nota obtenida en el bloque, superados los requisitos mínimos

*: Obtendrá una calificación de "NP" en la convocatoria ordinaria quien no tenga superado ningún bloque por evaluación continua y no se presente a ninguna prueba realizada durante el periodo de exámenes de esa convocatoria.

En las convocatorias extraordinarias obtendrá una calificación de "NP" quien no se presente a ninguna prueba en la convocatoria.

Sistema de revisión y comentario de exámenes

- El día de cada examen o prueba final de evaluación será anunciada la fecha de publicación de las notas, así como la fecha de revisión del examen o prueba.

- Se podrán comentar y revisar los resultados de las pruebas en las fechas previstas de acuerdo con la normativa vigente.

Bibliografía (básica y complementaria)

En el aula de la asignatura en el Campus Virtual de la UEx se encontrarán más recursos y referencias actualizados.

Bibliografía básica

- Walter Savitch, Resolución de problemas con C++. 5ª edición. Pearson-Addison Wesley, 2006
- Fatos Xhafa y otros, Programación en C++ para ingenieros. Thomson, 2006
- Jesús J. García Molina y otros. Una introducción a la programación. Un enfoque algorítmico. Thomson-Paraninfo, 2005

Bibliografía adicional

- H.M. Deitel y P.J. Deitel, C++. Cómo programar. 6ª edición. Pearson - Prentice Hall, 2008
- Luis Joyanes, Fundamentos de programación. Algoritmos, estructuras de datos y objetos. 4ª edición. McGraw-Hill, 2008
- Luis Joyanes, Fundamentos de programación. Libro de problemas. 2ª edición. McGraw-Hill, 2002
- J. Castro y otros, Curso de programación. McGraw-Hill, 93
- Luis Joyanes, Programación en C++. Algoritmos, estructuras de datos y objetos. McGraw-Hill, 2000

Otros recursos y materiales docentes complementarios

Medios materiales utilizados

- Pizarra
- Cañón de vídeo
- Ordenador

Materiales y recursos utilizados

Los materiales y recursos utilizados están en versión electrónica en el aula virtual de la asignatura:

- Transparencias o notas de clase para cada tema del programa
- Guiones de las sesiones de laboratorio
- Agenda del curso
- Hojas de ejercicios, enunciados y problemas resueltos

Algunos recursos y actividades del aula virtual son:

- Foros de comunicación
- Tablón de anuncios y novedades
- Recopilación de código fuente de programas
- Conjunto de referencias web relacionadas con la programación
- Vídeos explicativos
- Tests de autoevaluación de contenidos
- Problemas de autoevaluación
- Baterías de preguntas de test
- Tareas virtuales para la entrega de problemas

Tutorías de libre acceso

Las tutorías del profesorado de la asignatura están publicadas en sus respectivos despachos, en el aula virtual de la asignatura y en la web del centro.

Se pueden concertar reuniones fuera de los horarios fijados.

También se resolverán dudas a través de los espacios de comunicación del aula virtual.