

## PROGRAMA DE LA ASIGNATURA

**Curso académico: 2013-2014**

Identificación y características de la asignatura				
Código			Créditos ECTS	6
Denominación	Introducción a la programación			
Denominación en inglés	Introduction to programming			
Titulaciones	Grado en Ingeniería Informática en Ingeniería de Computadores Grado en Ingeniería Informática en Ingeniería del Software			
Centro	Escuela Politécnica			
Semestre	1	Carácter	Formación básica	
Módulo	Formación básica			
Materia	Informática			
Profesor/es				
Nombre	Despacho	Correo-e	Página web	
Alberto Gómez Mancha	12	agomez@unex.es		
M <sup>a</sup> Ángeles Mariscal Araujo	10	mariscal@unex.es		
Fernando Pulido Galán	10	fpgalan@unex.es		
Cristina Vicente Chicote	42	cristinav@unex.es		
Área de conocimiento	Lenguajes y Sistemas Informáticos			
Departamento	Ingeniería de Sistemas Informáticos y Telemáticos			
Profesor coordinador (si hay más de uno)	Alberto Gómez Mancha			
Competencias				
Competencias básicas (comunes a todas las enseñanzas de Grado)				
Se recogen por defecto				
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 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.

#### Resultados de aprendizaje asociados a estas competencias:

- 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.
- Comprende los conceptos básicos de la programación imperativa y puede aplicarlos en un problema de manera conveniente.
- Conoce y utiliza los tipos abstractos de datos fundamentales, y puede implementarlos en un lenguaje de alto nivel imperativo.
- Diseña, implementa, prueba y documenta programas de pequeño tamaño atendiendo a criterios de eficacia y legibilidad

### 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.

#### Resultados de aprendizaje de estas competencias:

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

### Objetivos de aprendizaje de la asignatura

Para desarrollar convenientemente las competencias asignadas a esta asignatura y poder alcanzar los resultados de aprendizaje propuestos, se establecen los siguientes objetivos de aprendizaje concretos, clasificados, según la taxonomía de Bloom, en los niveles de conocimiento, comprensión, aplicación y análisis.

#### Conocimiento

- Ob 1. Conocer las metodologías, técnicas y herramientas empleadas en el diseño y desarrollo de programas con el paradigma imperativo (CFB03)
- Ob 2. Definir los principios fundamentales de la programación estructurada (CFB04)
- Ob 3. Describir las características de las estructuras de datos en memoria interna básicas (CFB04)
- Ob 4. Enumerar las técnicas de prueba de programas (CFB04, CT14)
- Ob 5. Identificar técnicas básicas de trabajo en equipo (CT09)
- Ob 6. Nombrar las características básicas de los procesos de mejora de la calidad (CT14)

#### Comprensión

- Ob 7. Interpretar y reutilizar código que hayan sido escritos por otros desarrolladores (CFB03, CFB04)
- Ob 8. Explicar por escrito el proceso de desarrollo software de manera que, tanto el usuario de la aplicación, como otros desarrolladores, sean capaces de entender la solución propuesta (CFB04, CT09)
- Ob 9. Usar correctamente las estructuras de datos y algoritmos básicos según el problema planteado (CFB04)

- Ob 10. Plantear juegos de pruebas para la verificación de programas (CFB03, CT14)
- Ob 11. Calcular la complejidad de un algoritmo (CFB03, CT14)
- Ob 12. Determinar las estructuras de datos necesarias en un problema, así como los tipos abstractos de datos más adecuados (CFB04)
- Ob 13. Determinar los requisitos que un algoritmo necesita para su correcta implementación (CFB03, CT14)
- Ob 14. Planificar y documentar las reuniones realizadas en equipo (CT09)
- Ob 15. Fundamentar la importancia del aseguramiento de la calidad en el desarrollo de software (CFB03, CT14)

**Aplicación**

- Ob 16. Definir la estructura de datos adecuada para el almacenamiento de información (CFB04)
- Ob 17. Implementar los algoritmos que manejen las estructuras de datos definidas para el almacenamiento de los datos de un programa (CFB03, CFB04)
- Ob 18. Aplicar correctamente una metodología de desarrollo de software en la construcción de un programa utilizando el paradigma estructurado (CFB04, CT14)
- Ob 19. Aportar soluciones óptimas utilizando las herramientas conocidas (CFB03, CT14)
- Ob 20. Depurar adecuadamente los programas, utilizando diferentes herramientas (CFB03, CT14)
- Ob 21. Interaccionar correctamente con el resto de personas del grupo para alcanzar un objetivo común (CT09, CT14)
- Ob 22. Aplicar técnicas básicas de trabajo en equipo para alcanzar acuerdos en grupo y desarrollar actividades en grupos pequeños (CT09)
- Ob 23. Utilizar técnicas que permitan evaluar y mejorar la calidad de los sistemas software (CFB04, CT14)

**Análisis**

- Ob 24. Verificar que las soluciones a un problema cumplen con los objetivos propuestos y que realizan las tareas de forma eficaz (CFB04, CT14)
- Ob 25. Comparar distintas versiones de un algoritmo, eligiendo el más eficiente (CFB04, CT14)

<b>Objetivos de aprendizaje</b>	<b>Competencias</b>			
	CFB03	CFB04	CT09	CT14
<b>Conocimiento</b>				
Ob 1	X			
Ob 2		X		
Ob 3		X		
Ob 4		X		X
Ob 5			X	
Ob 6				X
<b>Comprensión</b>				
Ob 7	X	X		
Ob 8		X	X	
Ob 9		X		
Ob 10	X			X
Ob 11	X			X
Ob 12		X		
Ob 13	X			X
Ob 14			X	
Ob 15	X			X
<b>Aplicación</b>				
Ob 16		X		
Ob 17	X	X		
Ob 18		X		X
Ob 19	X			X

Ob 20	X			X
Ob 21			X	X
Ob 22			X	
Ob 23		X		X
<b>Análisis</b>				
Ob 24		X		X
Ob 25		X		X

## Temas y contenidos

### Breve descripción de contenidos

Fundamentos de programación y de estructuras de datos básicas, aplicaciones y propiedades. Características principales del paradigmas imperativo. Diseño, implementación y uso de tipos abstractos de datos. Lenguajes estructurados para el desarrollo de sistemas software.

### Temario de la asignatura

#### **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 y Conjunto

**Tema 10: Gestión dinámica de memoria**

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

**Actividades formativas**

Horas de trabajo del alumno por tema		Presencial		Actividad de seguimiento	No presencial
Tema	Total	GG	SL	TP	EP
1	3	1	0	0	2
2	7	1	2	0	4
3	13	3	2	0	8
4	26	6	4	0	16
5	10,5	1,5	1	0	8
6	7	2	1	0	4
7	9	1	2	0	6
8	9	1	2	0	6
9	33	11	4	0	18
10	28,5	7	3,5	0	18
Evaluación del conjunto	4	3	1	0	0
<b>TOTAL</b>	<b>150</b>	<b>37,5</b>	<b>22,5</b>	<b>0</b>	<b>90</b>

GG: Grupo Grande (100 estudiantes).

SL: Seminario/Laboratorio (prácticas en laboratorio; 15 estudiantes).

TP: Tutorías programadas (seguimiento docente).

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

**Actividades formativas que se plantearán**

A continuación se nombran algunas de las actividades formativas que se plantearán a lo largo del curso para alcanzar los objetivos 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 webs
- Visualización de videos de funcionamiento de herramientas
- Construcción colaborativa de glosarios en el aula virtual
- Comunicación con profesores y compañeros mediante foros
- Cuestionarios de evaluación y autoevaluación del trabajo en grupo

## **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:

- Portafolio de 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.

### **Portafolio de actividades**

El portafolio de actividades del estudiante está formado por distintas actividades simples realizadas por el 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 compañeros, 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 que tanto el estudiante como el profesor vean todo el trabajo realizado a lo largo del curso, reflexionen sobre la evolución y puedan aplicar las medidas correctoras necesarias para mejorar y alcanzar el nivel deseado al final 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 un futuro 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 objetivos de aprendizaje**

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

<b>Objetivos de aprendizaje</b>	<b>Instrumentos de evaluación</b>		
	<b>Portafolio (30%)</b>	<b>Proyecto (30%)</b>	<b>Pruebas (40%)</b>
<b>Conocimiento</b>			
Ob 1	X		X
Ob 2	X		
Ob 3	X		X
Ob 4	X	X	
Ob 5	X	X	
Ob 6	X	X	
<b>Comprensión</b>			
Ob 7	X	X	
Ob 8		X	
Ob 9	X	X	X
Ob 10	X	X	
Ob 11	X	X	X
Ob 12		X	X
Ob 13	X	X	X
Ob 14	X	X	
Ob 15	X		
<b>Aplicación</b>			
Ob 16	X	X	X
Ob 17	X	X	X
Ob 18		X	
Ob 19	X	X	X
Ob 20	X	X	
Ob 21	X	X	
Ob 22	X	X	

Ob 23		X	
<b>Análisis</b>			
Ob 24	X	X	X
Ob 25	X		X

### **Criterios de evaluación**

- Para superar esta asignatura deben superarse los requisitos mínimos de los 3 bloques de la asignatura (Portafolio, Proyecto y Pruebas), asociados a los tres instrumentos de evaluación principales (portafolio del estudiante, 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 de ese curso, siempre que el estudiante tenga derecho a examen en la convocatoria que supere el bloque.

### **Bloque 1: Portafolio**

- La nota del bloque de portafolio de actividades, NPOR, representa el 30% de la nota final.
- Este bloque se superará por evaluación continua 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 mediante el aula virtual.
- La asistencia a las clases de teoría y a las de laboratorio podrá considerarse, a efectos de evaluación, como actividades.
- Sólo se contabilizarán aquellas actividades que superen una calificación mínima.
- No es necesario obtener una nota mínima en este bloque para considerarlo superado.
- La nota de este bloque se obtiene a partir de la ponderación de las calificaciones obtenidas en las actividades propuestas.

### **Bloque 2: Proyecto**

- La nota del bloque de proyecto, NPRO, representa el 30% de la nota.
- Es obligatorio superar este bloque con una nota mínima de 5 sobre 10.
- Son requisitos indispensables para superar este bloque: entregar el programa y la documentación solicitados cumpliendo los requisitos mínimos que se establezcan, superar el examen de modificación propuesto y utilizar adecuadamente el sistema de desarrollo.
- La nota de este bloque es recuperable en las convocatorias oficiales de la asignatura, para lo que será necesario presentar el proyecto y superar las pruebas de modificación.

### **Bloque 3: Pruebas**

- La nota del bloque de pruebas escritas, NPRU, representa el 40% de la nota.
- Para superar este bloque es necesario obtener una nota mínima de 5 sobre 10.
- Este bloque podrá superarse por evaluación continua si se superan cada una de las pruebas parciales que se plantearán a lo largo del curso.
- La nota de este bloque es recuperable en las convocatorias oficiales mediante una prueba final. En la convocatoria ordinaria se podrá recuperar de forma independiente cada uno de los parciales. En el resto de convocatorias se realizará una prueba global de la materia.
- Cada prueba, parcial o final, podrá estar compuesto de preguntas de test o de respuestas cortas y resolución de problemas, con requisitos adicionales sobre la nota mínima que debe obtenerse en cada prueba para poder hacer media.



### Cálculo de la nota final de la asignatura

- La copia o el plagio en cualquier actividad o prueba supone una nota final de SUSPENSO (0) en la convocatoria y una nota de 0 en todas las calificaciones obtenidas hasta el momento para todos los implicados, además de las actuaciones legales pertinentes.
- Si se cumplen todos los requisitos mínimos de los tres bloques, la nota final se calcula como la siguiente media ponderada:  

$$\text{NotaFinal} \leftarrow 0,30 \text{ NPOR} + 0,30 \text{ NPRO} + 0,40 \text{ NPRU}$$
- Si no se cumplen todos los requisitos mínimos, la nota final será 2 ó 3, dependiendo de los casos.

En la siguiente tabla se puede consultar un tabla para 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	NCR/NP	Nota
<b>Nota final *</b>	<b>NP</b>	<b>2</b>	<b>3</b>	<b>2</b>	<b>3</b>	<b>3</b>	<b>Nota</b>

NP: no presentado a ese bloque

NCR: no se cumplen los requisitos mínimos de ese bloque

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

\*: La calificación será NP cuando no se haya realizado ningún esfuerzo apreciable para superar la asignatura.

### 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.
- El alumno podrá comentar y revisar sus resultados en las fechas previstas de acuerdo a la normativa vigente, para los exámenes de convocatorias oficiales.
- Para el resto de pruebas, la revisión se realizará en horario de clase o en el horario de tutorías de los profesores.

### Bibliografía y otros recursos

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

##### 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 y en algunos casos también en papel, disponibles en el servicio de reprografía:

- Transparencias para cada tema del programa

- Guiones de las sesiones de laboratorio
- Agenda del curso
- Con recursos propios del aula virtual los siguientes:
  - Sistemas de participación
    - Foros de comunicación
    - Tablón de anuncios y novedades
  - Información adicional
    - Glosarios de términos y palabras claves
    - Recopilación de código fuente de programas
    - Conjunto de referencias web relacionadas con la programación
    - Vídeos explicativos
  - Autoevaluación
    - Tests de autoevaluación de contenidos
    - Problemas de autoevaluación
    - Baterías de preguntas de test
- Tareas virtuales para la entrega de problemas

### Horario de tutorías

Tutorías programadas: no hay en esta asignatura

Tutorías de libre acceso:

Las tutorías de cada profesor 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 horario fijados.

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

### Recomendaciones

“Introducción a la programación” es una asignatura fundamental, en la que se basan muchas asignaturas posteriores del plan de estudios.

Recomendaciones para el desarrollo adecuado de la asignatura:

- Seguir el plan de trabajo marcado, prestando especial atención a la resolución de problemas implementados en un lenguaje de programación y ejecutados en el ordenador.
- Asistir a clase, tanto de teoría como de laboratorio, y realizar las tareas necesarias para el seguimiento de las mismas, en tiempo y forma.
- Acceder de manera regular y continuada al aula virtual de la asignatura, participar activamente en los foros y realizar las actividades propuestas durante el curso.
- Acudir a las tutorías del profesorado para resolver las dudas.

Normas de la asignatura:

- Durante el mes de septiembre se deberá rellenar el perfil del estudiante en el aula virtual, incluyendo fotografía clara y actualizada.
- En los mensajes de correo electrónico enviados a los profesores siempre deberá constar el nombre completo y la asignatura del emisor.
- Cualquier comunicación escrita debe realizarse siguiendo las normas básicas de ortografía y redacción, y siempre mostrando el máximo respeto por los demás.
- La utilización del ordenador y del acceso a Internet en las sesiones de laboratorio para fines no relacionados directamente con la asignatura no está permitido
- Es responsabilidad del estudiante la custodia y protección de sus programas y trabajos. No se deben realizar trabajos en grupo a no ser que esté expresamente indicado en la descripción del mismo.
- De acuerdo a la normativa del Centro, aquellos estudiantes a los que les coincida la fecha del examen oficial con el examen de otra asignatura deberán solicitar el cambio de fecha del examen mediante instancia presentada en la secretaría del centro, en el período establecido por la misma. NO se realizará ningún examen extra si no se ha seguido el procedimiento oficial establecido.
- **Cualquier duda, caso no contemplado en este programa o problema debe ser consultado con suficiente antelación con los profesores de la asignatura.**