

Programa de la asignatura ESTRUCTURAS DE DATOS Y DE LA INFORMACIÓN

Curso académico: 2013/2014

Identificación y características de la asignatura				
Código	501271			Créditos ECTS 6
Denominación	Estructuras de datos y de la información			
Denominación en inglés	Data structures and algorithms			
Titulaciones	Grado en Ingeniería Informática en Ingeniería de Computadores Grado en Ingeniería Informática en Ingeniería del Software			
Centro	Escuela Politécnica			
Semestre	2º	Carácter	Formación básica	
Módulo	Formación básica			
Materia	Informática			
Profesor/es				
Nombre	Despacho	Correo-e	Página web	
Julia González Rodríguez	13	juliagon@unex.es	http://epcc.unex.es	
M ^a Ángeles Mariscal Araújo	10	mariscal@unex.es	http://epcc.unex.es	
J. Antonio Rico Gallego	37	jarico@unex.es	http://epcc.unex.es	
Encarna Sosa Sánchez	10	esosa@unex.es	http://epcc.unex.es	
J. María Conejero Manzano	20	chemacm@unex.es	http://epcc.unex.es	
Área de conocimiento	Lenguajes y sistemas informáticos			
Departamento	Ingeniería en sistemas informáticos y telemáticos			
Profesor coordinador	Julia González Rodríguez			
Competencias				
Competencias básicas				
CB02: Que los estudiantes sepan aplicar sus conocimientos a su trabajo 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.				
Competencias específicas de formación básica				
Según los planes de estudio aprobados, esta asignatura debe cubrir, total o 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 de estas competencias:				
<ul style="list-style-type: none"> 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 la programación orientada a objetos, diferencia ambos paradigmas 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 u orientado a objetos.
- Diseña, implementa, prueba y documenta programas de pequeño tamaño atendiendo a criterios de eficacia y legibilidad

Competencias transversales asignadas

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.

CT02: Habilidades de gestión de recursos de información.

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

Resultados de aprendizaje de estas competencias:

- Conoce y utiliza los recursos de información actuales (bibliotecas e internet) adecuadamente.
- Encuentra información específica entre diferentes fuentes de información, discriminando adecuadamente su idoneidad y fiabilidad y es capaz de integrarlos en el entorno de su trabajo.
- Aplica correctamente la incorporación de referencias en la elaboración de proyectos, informes, documentos, etc.
- Conoce las características básicas de un sistema de garantía de calidad y de los sistemas de mejora continua de la calidad.
- Aplica sus conocimientos técnicos para planificar actuaciones con el objetivo de asegurar y mejorar la calidad de los sistemas.

Objetivos de aprendizaje de la asignatura

Para desarrollar convenientemente las competencias asignadas a esta asignatura y poder alcanzar 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

- Obj 1. Conocer las metodologías, técnicas y herramientas empleadas en el diseño y desarrollo de programas, fundamentalmente orientado a objetos. (CFB3)
- Obj 2. Definir los principios fundamentales de la programación orientada a objetos (CFB4)
- Obj 3. Enunciar las estructuras de datos más habituales (CFB4)
- Obj 4. Conocer las técnicas de prueba de programas (CFB4, CT14)
- Obj 5. Conocer métodos que permitan estimar la calidad de un algoritmo (CB02, CFB3, CT14)
- Obj 6. Conocer las principales fuentes de información relacionadas con la programación y la resolución de algoritmos (CT2)

Comprensión

- Obj 7. Comprender y reutilizar código que hayan sido escritos previamente por otros desarrolladores (CFB3, CFB4, CT2)
- Obj 8. Describir 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, referenciando las fuentes utilizadas (CFB4, CT2)
- Obj 9. Identificar las estructuras de datos necesarias en un problema, así como la jerarquía de clases más adecuada (CFB4)
- Obj 10. Usar correctamente las estructuras de datos y algoritmos básicos según el problema planteado (CFB4)
- Obj 11. Determinar los requisitos de un algoritmo para su correcta implementación y su ejecución eficiente. (CFB3, CT14)
- Obj 12. Usar juegos de pruebas para la verificación de programas. (CFB3, CT14)
- Obj 13. Calcular la complejidad de un algoritmo (CFB3, CT14)

Aplicación

- Obj 14. Construir la estructura de datos adecuada para el almacenamiento de los datos del problema a utilizar (CFB4)
- Obj 15. Definición de los algoritmos que manejen las estructuras de datos del un programa (CFB3, CFB4)
- Obj 16. Aplicar correctamente una metodología de desarrollo de software orientado a objetos en la construcción de un programa utilizando el paradigma orientado a objetos (CFB4, CT2, CT14)
- Obj 17. Aportar soluciones óptimas al problema planteado utilizando las herramientas aprendidas y recursos bibliográficos adecuados. (CB02, CFB3, CT2, CT14)
- Obj 18. Depurar adecuadamente los programas, utilizando diferentes herramientas. (CFB3, CT14)
- Obj 19. Incluir software previamente desarrollado como parte de la solución al problema propuesto (CB02, CFB4, CT2)

Análisis

- Obj 20. Comparar distintas versiones de un algoritmo, eligiendo el más eficiente (CFB4, CT14)
- Obj 21. Verificar que las soluciones a un problema cumplen con los objetivos propuestos y que realizan las tareas de forma eficaz (CB02, CFB4, CT14)

Tabla resumen de relación entre competencias y objetivos:

Objetivos de aprendizaje		CB02	CFB03	CFB04	CT02	CT14
Conocimiento	Obj. 01		X			
	Obj. 02			X		
	Obj. 03			X		
	Obj. 04			X		X
	Obj. 05	X	X			X
	Obj. 06				X	
Comprensión	Obj. 07		X	X	X	
	Obj. 08			X	X	
	Obj. 09			X		
	Obj. 10			X		
	Obj. 11		X			X
	Obj. 12		X			X
	Obj. 13		X			X
Aplicación	Obj. 14			X		
	Obj. 15		X	X		
	Obj. 16			X	X	X
	Obj. 17	X	X		X	X
	Obj. 18		X			X
	Obj. 19	X		X	X	
Análisis	Obj. 20			X		X
	Obj. 21	X		X		X

Temas y contenidos

Breve descripción del contenido

Fundamentos de programación y de estructuras de datos básicas, aplicaciones y propiedades. Características principales del paradigma orientado a objetos. Lenguajes orientados a objetos para el desarrollo de sistemas software.

Temario de la asignatura

<p>Tema 1: Programación orientada a objetos</p> <p>1.1 Conceptos básicos, descripción y terminología fundamental</p> <p>1.2 Principios fundamentales de POO</p> <p>1.3 Especificación, implementación y uso de clases</p> <p>1.4 Operaciones fundamentales con instancias (comparación, copia, sobrecarga,...)</p>

<p>Tema 2: Análisis y diseño de sistemas orientado a objetos</p> <p>2.1 Conceptos básicos, descripción y terminología fundamental</p> <p>2.2 Introducción al desarrollo del software orientado a objetos</p> <p>2.3 Modelo conceptual</p> <p>2.4 Diagrama de clases</p>
--

<p>Tema 3: Estructuras de datos lineales.</p> <p>3.1 Conceptos básicos, descripción y terminología fundamental</p> <p>3.2 Pilas</p> <p>3.3 Listas</p> <p>3.4 Colas</p> <p>3.5 Aplicaciones</p>

<p>Tema 4: Estructuras de almacenamiento secundario</p> <p>4.1 Conceptos básicos, descripción y terminología fundamental</p> <p>4.2 Ficheros de texto</p> <p>4.3 Ficheros binarios</p> <p>4.4 Algoritmos fundamentales del tratamiento de ficheros</p>

<p>Tema 5: Estructuras de datos no lineales</p> <p>5.1 Conceptos básicos, descripción y terminología fundamental</p> <p>5.2 Especificación del árbol binario y su uso</p> <p>5.3 Recorridos de árboles binarios</p> <p>5.4 Aplicaciones de árboles binarios</p> <p>5.5 Árboles generales</p>

Actividades formativas

Horas de trabajo del alumno por tema		Presencial		Actividad de seguimiento	No presencial
		GG	SL	TP	EP
Tema	Total				
1	29	9	6	0	14
2	15	3	2	0	10
3	38	12	6	0	20
4	18	3	3	0	12
5	31	7	4	0	20
Evaluación	19	3,5	1,5	0	14
Total	150	37,5	22,5	0	90

GG: Grupo Grande (hasta 100 estudiantes).
 SL: Seminario/Laboratorio (prácticas laboratorio o campo = hasta 15 estudiantes).
 TP: Tutorías Programadas (seguimiento docente, tipo tutorías ECTS).
 EP: Estudio personal, trabajos individuales o en grupo, y lectura de bibliografía.

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:

- Carpeta 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.

Carpeta de actividades

La carpeta de actividades del estudiante está formada 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 la carpeta de actividades como herramienta de evaluación, la carpeta 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, los problemas presentados se adecuarán 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.

Objetivos de aprendizaje	Instrumentos de evaluación		
	Carpeta-20%	Proyecto-40%	Pruebas-40%
Conocimiento			
Obj 01	X		X
Obj 02	X		X
Obj 03	X		X
Obj 04	X		X
Obj 05	X		X
Obj 06			X
Comprensión			
Obj 07	X		X
Obj 08		X	
Obj 09	X	X	X
Obj 10	X	X	X
Obj 11	X	X	X
Obj 12		X	
Obj 13	X	X	X
Aplicación			
Obj 14	X	X	X
Obj 15	X	X	X
Obj 16		X	
Obj 17	X	X	X

Obj 18		X	
Obj 19		X	
Análisis			
Obj 20	X	X	
Obj 21	X	X	

Criterios de evaluación

- Para superar esta asignatura deben superarse los requisitos mínimos de los 3 bloques de la asignatura (Carpeta, Proyecto y Pruebas), asociados a los tres instrumentos de evaluación principales (carpeta 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 las convocatorias de JUNIO y JULIO del presente curso y durante la convocatoria extraordinaria de FEBRERO del siguiente, siempre que el estudiante tenga derecho a examen en la convocatoria que supera el bloque y en el que se pone la nota. Aquellos alumnos que adelanten la convocatoria de JUNIO a FEBRERO no tendrán derecho a realizar el examen o defensa en la convocatoria de JUNIO, por lo tanto no se les podrá guardar la nota de los bloques de teoría o proyecto, aunque sean aprobados por parciales o fases.

Bloque 1: Carpeta de actividades

- La nota del bloque de carpeta de actividades, **NCARP**, representa el **20%** de la nota final de la asignatura.
- Este bloque se superará exclusivamente por **evaluación continua** realizando las actividades propuestas en el aula virtual (foros, cuestionarios,...), hojas de clase y sesiones de laboratorio.
- Cada actividad será calificada en la escala: mal, insuficiente, suficiente, bien y muy bien.
- No es necesario obtener una nota mínima en este bloque para considerarlo superado.
- La nota de este bloque se obtiene sumando las calificaciones obtenidas en las actividades propuestas.
- La nota de este bloque **no es recuperable** mediante ningún examen posterior, en ninguna convocatoria ni oficial ni extraordinaria.

Bloque 2: Proyecto

- La nota del bloque de proyecto, **NPRO**, representa el **40%** de la nota final de la asignatura.
- El proyecto se realizará en fases.
- De cada una de las fases se realizará una defensa.
- Para superar el bloque proyecto es necesario superar la defensa de cada una de las fases y obtener una puntuación de 5 sobre 10 en la nota global.
- Se realizarán dos fases:
 - o La primera fase se entregará en el mes de abril.
 - o La segunda fase se entregará en la convocatoria oficial de junio.
- El cálculo de la nota de este bloque, aprobadas las defensas de las fases, se calculará como:

$$\mathbf{NPRO \leftarrow Fase02}$$

Proyecto en convocatorias oficiales

- o Convocatoria de Junio, existirán dos posibilidades:
 1. Entrega del proyecto completo y defensa de las fases 01 y 02 para aquellos estudiantes que no hayan superado o no hayan entregado la fase 01.
 2. Entrega del proyecto completo y defensa de la fase 02, para aquellos estudiantes que hayan superado la fase 01 del proyecto.
- o Resto de convocatorias:
 1. Entrega y defensa (de las dos fases) del proyecto completo.
- Son requisitos indispensables para superar este bloque: entregar los programas y su documentación solicitados cumpliendo los requisitos mínimos que se establezcan, entre ellos gestionar adecuadamente la memoria dinámica utilizada y superar los exámenes de modificación propuestos y utilizar adecuadamente el sistema de desarrollo.
- La nota de este bloque **es recuperable** en las convocatorias oficiales de la asignatura en las que el estudiante esté matriculado, para lo que será necesario presentar los programas y la documentación del mismo de acuerdo al enunciado propuesto para la convocatoria y además

superar las pruebas de modificación y defensa del proyecto.

- En la convocatoria de JUNIO se podrán realizar ampliaciones del proyecto, que una vez superados los requisitos del mismo podrán contribuir a mejorar la nota obtenida hasta en 2 puntos extras, en este caso la nota podrá llegar a ser de 12 puntos.

Bloque 3: Pruebas escritas

- La nota del bloque de pruebas escritas, **NPRU**, representa el **40%** de la nota final de la asignatura.
- Para superar este bloque es necesario obtener una nota mínima global de 5 sobre 10.
- La nota de este bloque **es recuperable** en las convocatorias oficiales de la asignatura en las que el estudiante esté matriculado.

- Superación por parciales

- o Son requisitos indispensables para superar este bloque superar cada una de las pruebas parciales que se plantearán a lo largo del curso (Obtener un 5 sobre 10 como resultado final de cada prueba parcial).
- o Se realizarán dos pruebas:
 - La primera prueba se realizará en el mes de abril.
 - La segunda prueba se realizará el mismo día que el examen de la convocatoria oficial de junio.
- o El cálculo de la nota de este bloque, cuando se apruebe por parciales se calculará como:

$$\text{NPRU} \leftarrow 0,60 \text{ Prueba01} + 0,40 \text{ Prueba02}$$

- Prueba final en convocatorias oficiales
 - o Convocatoria de Junio, se realizarán dos pruebas diferenciadas:
 1. Aquellos estudiantes que hayan superado la prueba 01, sólo tendrán que realizar una prueba correspondiente a la prueba 02.
 2. Aquellos estudiantes que no han superado o no hayan realizado el examen parcial 01, deberán superar este bloque con la realización de un examen que incluirá todos los contenidos de la asignatura.
 - o Resto de convocatorias:
 - El examen será único para todos los estudiantes.
- Cada prueba estará formada por un conjunto de ejercicios en los que será necesario obtener una nota igual o superior de 4 sobre 10 en cada uno para hacer media y obtener la calificación de la prueba, sea esta parcial o final.
- El incumplimiento de uno de los requisitos mínimos (obtener un 4 sobre 10) en un ejercicio de una prueba, supone que la prueba no está superada.

Cálculo de la nota final de la asignatura

- La copia o el plagio o el uso de sistemas o información no autorizada 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 indicadas según la normativa vigente.
- Si se cumplen todos los requisitos mínimos de los tres bloques, la nota final se calcula como la siguiente media ponderada:

$$\text{NotaFinal} \leftarrow 0,20 \text{ NCARP} + 0,40 \text{ NPRO} + 0,40 \text{ NPRU}$$

- Aquellos casos en los que no se obtiene una calificación de No presentado ni 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 Escritas	NP	NCR	NP	Nota	NCR	Nota
Proyecto	NP	NP	NCR	NCR	Nota	Nota
Nota final*	NP	2		3		Media

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

- Se obtendrá una calificación final de **No Presentado** cuando
 - En la convocatoria de JUNIO:
 - En el bloque de teoría: se haya obtenido una calificación de NO PRESENTADO en todas las pruebas (parciales y finales).
 - En el bloque de proyecto: se haya obtenido una calificación de NO PRESENTADO en todas las fases.
 - En el resto de convocatorias:
 - En el bloque de teoría: se haya obtenido una calificación de NO PRESENTADO en el examen de la convocatoria.
 - En el bloque de proyecto: se haya obtenido una calificación de NO PRESENTADO en la defensa del proyecto.

Sistema de revisión y comentario de exámenes

- El estudiante 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 clases o en el horario de tutorías de los profesores.

Bibliografía y otros recursos

Bibliografía básica

- "Estructuras de Datos y Algoritmos". Roberto Hernández, Juan Carlos Lázaro, Raquel Dormido, Salvador Ros. Universidad Nacional de Educación a Distancia , Ed. Prentice Hall, 2000.
- "Construcción de Software Orientada a Objetos. 2ª Edición", Bertrand Meyer, Ed. Prentice Hall, 1999.
- "Programación Orientada a Objetos". Roberto Rodríguez Echeverría, Encarna Sosa Sánchez y Álvaro Prieto Ramos. Editado por Librería Álvaro (Cáceres). 2004.
- "Resolución de problemas con C++. 5ª edición". Walter Savitch. Prentice Hall, 2007.

Bibliografía adicional

- "Fundamentos de programación. Algoritmos, estructuras de datos y objetos. 4ª edición". Luis Joyanes. Ed. McGraw-Hill, 2008
- "Fundamentos de programación. Libro de problemas. 2ª edición". Luis Joyanes, Ed. McGraw-Hill, 2003
- "Object-Oriented Analysis & Design". McLaughlin, Pollice and West. Head First. Ed. O'reilly. 2006
- Aprende C++ como si estuvieras en primero:
<http://www.tecnun.es/asignaturas/Informat1/AyudaInf/aprendainf/cpp/avanzado/cppavan.pdf>

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:

- Transparencias y material para cada tema del programa
- Guiones de las sesiones de laboratorio
- Proyectos de programación (completos y de trabajo)
- Lista de problemas
- Agenda del curso

Son 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 existen

Tutorías de libre acceso:

- Julia Gonzalez: martes y miércoles y jueves de 10:00 a 11:30 y de 13:30 a 14:00.
- M^a Ángeles Mariscal: martes, miércoles y jueves de 9:30 a 11:30.
- Juan Antonio Rico Gallego: martes y viernes de 9:30 a 11:30, miércoles de 17:30 a 19:30.
- José María Conejero: lunes, martes y jueves de 11:30 a 13:30.
- Encarna Sosa: lunes de 11:30 a 13:30, martes de 9:30 a 11:30 y miércoles de 18:30 a 20:30.

Esta información podrá modificarse según la normativa vigente, por lo que la información actualizada estará disponible en el aula virtual, en la puerta de los despachos de los profesores y registrada según la normativa.

La comunicación entre profesor-estudiante y estudiante-estudiante será continua a lo largo de todo el curso mediante diferentes canales de comunicación electrónicos. Los utilizados en la asignatura serán: foros en el aula virtual, correo electrónico y redes sociales (el perfil de twitter de la asignatura es **@EDIUEx**).

Recomendaciones

- 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.
- El acceso regular y continuado al aula virtual de la asignatura, la participación activa en los foros y la realización de las actividades propuestas durante el curso.
- Acudir a las tutorías del profesorado para resolver las dudas.
- En caso de adelantar el examen a la convocatoria extraordinaria, adelantar la convocatoria de JULIO a FEBRERO.

Los estudiantes de las titulaciones Ingeniería en Informática, Ingeniería técnica en Informática de Sistemas e Ingeniería técnica en Informática de Gestión que hayan cursado y superado las asignaturas de Laboratorio de Programación I o Laboratorio de Programación II deben hablar con los profesores de esta asignatura.

Aquellos estudiantes que no hayan cursado la asignatura de Introducción a la Programación (IP), deben completar su formación con los contenidos allí tratados. En cualquier caso se recomienda cursar la asignatura IP antes de cursar la asignatura de Estructuras de datos y de la Información (EDI), pues los contenidos allí estudiados son requisito previo para el estudio de EDI.