

Guía Docente de la Asignatura: Estructuras de datos y algoritmos

Responsable	Prof. D. José Antonio Marcos					
Facultad	Ciencias y Tecnología					
Titulación	Grado en Ingeniería Informática					
Materia	Informática					
Plan	2012					
Carácter	Formación Básica					
Periodo de impartición	Trimestral					
Curso/es	Primero					
Nivel/Ciclo	Grado					
Créditos ECTS	Teóricos	6	Prácticos	0	Total	6
Lengua en la que se imparte	Castellano					
Datos de Contacto:	Correo electrónico: joseantonio.marcos@ui1.es					

Asignaturas de la Materia	Asignaturas	Carácter	Curso	Créditos	Horas
	Estructura y tecnología de computadores I.	FB	1º	6	150
	Laboratorio de programación.	FB	1º	6	150
	Diseño y programación orientada a objetos.	FB	2º	6	150
	Estructura y tecnología de computadores II.	FB	1º	6	150
	Estructuras de datos y algoritmos.	FB	1º	6	150
Contextualización curricular de la asignatura	<p>En las asignaturas de programación se persigue el objetivo de facilitar al alumno herramientas y mostrarle su uso para llegar a definir y ejecutar programas que apliquen determinados tipos de algoritmos. Dichos algoritmos resuelven, o intentan resolver, problemas de un cierto grado de complejidad. Todo ello partiendo de unas técnicas básicas dentro del ámbito de la programación, estructurada u orientada a objetos, y sin detenernos demasiado en contemplar aspectos como la eficiencia y el coste del algoritmo usado o la tipología de las estructuras de datos que se implementan.</p> <p>En esta asignatura tomaremos estos últimos apartados y realizaremos un análisis detallado de cada una de las estructuras de datos más comúnmente utilizadas, observando los aspectos comentados sobre eficiencia y coste y aplicándolos sobre los algoritmos diseñados.</p> <p>De esta manera se pretende concienciar al alumno que al afrontar el desarrollo y la implementación de un programa mediante el diseño del correspondiente algoritmo, debe tener muy presente los aspectos que aquí se tratan con la finalidad de conseguir aplicaciones más eficientes y ligeras, permitiendo un uso más eficaz y sostenible de las mismas.</p> <p>Objetivos:</p> <ul style="list-style-type: none"> • Comprender el apartado teórico con el fin de poder aplicar en el ámbito práctico la funcionalidad e implementación de los Tipos de Datos Abstractos. • Lograr calcular los recursos necesarios y los recursos utilizados en una implementación, tanto en su aspecto temporal como en su apartado espacial. • Comprender el desarrollo de programas y algoritmos, promoviendo la reutilización de código ya existente. • Comprender y aplicar correctamente las estructuras fundamentales analizadas. • Aplicar lo estudiado mediante el uso de un lenguaje algorítmico y, posteriormente, usando el lenguaje de programación C. 				
Prerrequisitos para cursar la asignatura	Ninguno.				

<p>Generales de la Materia</p>	<p>CB09: Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.</p> <p>CB10: Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.</p>	
<p>Propias de la Universidad</p>	<p>CU09: Considerar los valores propios de la Formación Profesional Superior en términos de igualdad formativa y educativa con la universitaria.</p>	<p>Transversales</p> <p>CT01: Capacidad de análisis y síntesis: encontrar, analizar, criticar (razonamiento crítico), relacionar, estructurar y sintetizar información proveniente de diversas fuentes, así como integrar ideas y conocimientos.</p> <p>CT04: Capacidad para la resolución de problemas.</p>
<p>Competencias de la Asignatura</p>	<p>CB09: Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.</p> <p>CB10: Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.</p> <p>CT01: Capacidad de análisis y síntesis: encontrar, analizar, criticar (razonamiento crítico), relacionar, estructurar y sintetizar información proveniente de diversas fuentes, así como integrar ideas y conocimientos.</p> <p>CT04: Capacidad para la resolución de problemas.</p>	

Actividades Formativas de la Materia	Trabajo dirigido		ECTS	HORAS	Trabajo autónomo del alumno		ECTS	HORAS	
	<i>Comunidad de aprendizaje (Aula Virtual)</i>					Actividades de trabajo autónomo individual (Estudio de la Lección).	8	200	
	Actividades de descubrimiento inducido (Estudio del Caso).	5,76	154	Actividades de aplicación práctica (individuales).	2,88	72			
	Actividades de Interacción y colaboración (Foros-Debates de apoyo al caso y a la lección).	1,92	48	Lectura crítica, análisis e investigación.	3,6	90			
	Actividades de aplicación práctica (grupal online).	0	0	Actividades de evaluación.	0,4	10			
	Presentaciones de trabajos y ejercicios.	0	0	<i>Prácticas externas.</i>	0	0			
	Seminarios.	0	0	<i>Prácticas de iniciación profesional.</i>	0	0			
	<i>Interacción alumno-tutor (Aula Virtual).</i>					Trabajo Fin de Grado	0	0	
	Tutorías.	0,32	8						
	Presentaciones de trabajos y ejercicios propuestos.	0,64	16						
	Actividades de evaluación.	0,48	12						
	Total	9,12	228	Total	14,88	372			

Actividad	Descripción
Trabajo dirigido.	
<i>Comunidad de aprendizaje (Aula Virtual).</i>	
Actividades de descubrimiento inducido (Estudio del Caso).	Actividades en las que el alumno podrá llevar a cabo un aprendizaje contextualizado trabajando, en el Aula Virtual y de manera colaborativa, una situación real o simulada que le permitirá realizar un primer acercamiento a los diferentes temas de estudio.
Actividades de Interacción y colaboración (Foros-Debates de apoyo al caso y a la lección).	Actividades en las que se discutirá y argumentará acerca de diferentes temas relacionados con las asignaturas de cada materia y que servirán para guiar el proceso de descubrimiento inducido.
Actividades de aplicación práctica (grupal online).	Incluye la resolución de problemas, elaboración de proyectos y actividades similares que permitan aplicar los aspectos conceptuales, procedimentales y actitudinales trabajados en otras partes de las asignaturas.

	Presentaciones de trabajos y ejercicios.	Incluye la elaboración conjunta en el Aula Virtual y, en su caso, defensa virtual de los trabajos y ejercicios solicitados conforme a los procedimientos de defensa que se establezcan en las guías docentes.
	Seminarios.	Incluye la asistencia presencial o virtual a sesiones en pequeño grupo dedicadas a temáticas específicas de cada asignatura.
	Interacción alumno-tutor (Aula Virtual).	
	Tutorías	Permiten la interacción directa entre docente y alumno para la resolución de dudas y el asesoramiento individualizado sobre distintos aspectos de las asignaturas.
	Presentaciones de trabajos y ejercicios propuestos.	Incluye la elaboración individual, presentación y, en su caso, defensa virtual de los trabajos y ejercicios solicitados, conforme a los procedimientos de defensa que se establezcan en las guías docentes.
	Actividades de evaluación.	<i>Véase información al respecto en el apartado siguiente.</i>
	Trabajo Autónomo del alumno	
	<i>Actividades de trabajo autónomo individual (Estudio de la Lección).</i>	Trabajo individual de los materiales utilizados en las asignaturas, aunque apoyado por la resolución de dudas y construcción de conocimiento a través de un foro habilitado para estos fines. Esta actividad será la base para el desarrollo de debates, resolución de problemas, etc.
	Actividades de aplicación práctica (individuales).	Incluye el trabajo individual en la resolución de problemas, elaboración de proyectos y actividades similares que permitan aplicar los aspectos conceptuales, procedimentales y actitudinales trabajados en otras partes de la asignatura.
	Lectura crítica, análisis e investigación.	Se trata de actividades en las que el alumno se acerca a los diferentes campos de estudio con una mirada crítica que le permite un acercamiento a la investigación. Se incluyen, a modo de ejemplo, recensiones de libros o crítica de artículos y proyectos de investigación.
Actividades de evaluación.	<i>Véase información al respecto en el apartado siguiente.</i>	
Actividades Formativas de la Asignatura	Actividad	Descripción
	Trabajo dirigido.	
	Comunidad de aprendizaje (Aula Virtual).	

Actividades de descubrimiento inducido (Estudio del Caso).	Actividades en las que el alumno podrá llevar a cabo un aprendizaje contextualizado trabajando, en el Aula Virtual y de manera colaborativa, una situación real o simulada que le permitirá realizar un primer acercamiento a los diferentes temas de estudio.
Actividades de Interacción y colaboración (Foros-Debates de apoyo al caso y a la lección).	Actividades en las que se discutirá y argumentará acerca de diferentes temas relacionados con las asignaturas de cada materia y que servirán para guiar el proceso de descubrimiento inducido.
Interacción alumno-tutor (Aula Virtual).	
Tutorías	Permiten la interacción directa entre docente y alumno para la resolución de dudas y el asesoramiento individualizado sobre distintos aspectos de las asignaturas.
Presentaciones de trabajos y ejercicios propuestos.	Incluye la elaboración individual, presentación y, en su caso, defensa virtual de los trabajos y ejercicios solicitados, conforme a los procedimientos de defensa que se establezcan en las guías docentes.
Actividades de evaluación.	El sistema de evaluación final será común para todas las asignaturas de la materia y se basará en una selección de las pruebas de evaluación más adecuadas para el tipo de competencias que se trabajen.
Trabajo Autónomo del alumno	
<i>Actividades de trabajo autónomo individual (Estudio de la Lección).</i>	Trabajo individual de los materiales utilizados en las asignaturas, aunque apoyado por la resolución de dudas y construcción de conocimiento a través de un foro habilitado para estos fines. Esta actividad será la base para el desarrollo de debates, resolución de problemas, etc.
Actividades de aplicación práctica (individuales).	Incluye el trabajo individual en la resolución de problemas, elaboración de proyectos y actividades similares que permitan aplicar los aspectos conceptuales, procedimentales y actitudinales trabajados en otras partes de la asignatura.
Lectura crítica, análisis e investigación.	Se trata de actividades en las que el alumno se acerca a los diferentes campos de estudio con una mirada crítica que le permite un acercamiento a la investigación. Se incluyen, a modo de ejemplo, reseñas de libros o crítica de artículos y proyectos de investigación.
Actividades de evaluación.	El sistema de evaluación final será común para todas las asignaturas de la materia y se basará en una selección de las pruebas de evaluación más adecuadas para el tipo de competencias que se trabajen.

<p>Proceso de Aprendizaje</p>	<p>Para cada Unidad Didáctica el alumno deberá llevar a cabo actividades para lograr afianzar conceptos y la aplicación práctica de estos, proponiéndose actividades como:</p> <ul style="list-style-type: none"> • Estudio de caso, exponiendo una situación, real o simulada, en la que se apliquen conceptos analizados en la Unidad Didáctica y deriven en la propuesta de resolución de algún ejercicio extraído del contexto descrito. • Contenidos teóricos: Conceptos teóricos, que se focalizan en los aspectos fundamentales y dan forma y estructuran el resto de contenido de cada una de las Unidades Didácticas. • Foros de Debate: Discusión de conceptos en foros, con el objetivo de tomar el pulso a los alumnos y comprobar el nivel de calado y consolidación de los contenidos estudiados en la Unidad Didáctica. • Trabajo colaborativo/individual: Trabajo individual en la resolución de problemas y ejercicios propuestos, fomentando el apartado de investigación e instrumentalización de las herramientas y conceptos mostrados en la Unidad Didáctica.
<p>Orientaciones al estudio</p>	<p>Siguiendo las directrices fundamentales establecidas por esta Universidad, en las que se toma como aspecto fundamental la búsqueda y el desarrollo de habilidades de pensamiento para conseguir que el alumno logre adquirir de forma autónoma nuevos conocimientos, proponemos la aplicación de los siguientes puntos:</p> <ul style="list-style-type: none"> • Estudio inicial de materia e información a través de una propuesta basada en resolución de problemáticas sencillas, para facilitar la focalización sobre los fundamentos teóricos y/o prácticos a analizar y asimilar por parte del alumno • Fomentar la comprensión y análisis de problemas y enunciados con el propósito de trasladar dicha información al entorno de la programación y el análisis algorítmico, buscando la resolución más óptima y eficiente de los mismos. • Encadenar los conceptos de estructuras de datos y desarrollo de algoritmos estudiados con sus aplicaciones en el entorno de la ingeniería informática. • Facilitar conjuntos y colecciones de ejercicios y ejemplos para proporcionar un amplio abanico de información que permita al alumno la puesta en práctica de la mecánica de resolución de los mismos y consolide conceptos e instrumentos de resolución. • Potenciar las relaciones existentes entre los conceptos teóricos y los procedimientos prácticos tratados a lo largo de todo el contenido de la asignatura e integrarlos modo de hilo conductor de la misma. • Potenciar el trabajo en equipo y los foros de debate para promover la dialéctica y la discusión entre los alumnos.
<p>Resultados de Aprendizaje de la Materia</p>	<p>Al completar de forma exitosa esta materia, el alumno:</p> <ul style="list-style-type: none"> • Comprende el funcionamiento de un computador, y la necesidad de desarrollo de software por parte del programador. • Conoce los distintos paradigmas de programación, situando en ese contexto el lenguaje de programación que se va a utilizar. • Comprende la necesidad de un proceso de traducción de un lenguaje de alto nivel. • Conoce y distingue los conceptos de algoritmo y programa. • Conoce los tipos de datos primitivos y sus operaciones. • Distingue entre tipo de dato y objeto. • Conoce las acciones básicas de E/S de datos. • Aprende a usar las estructuras de control básicas: secuencial, condicional e iterativa.

- Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado*.
- Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos*.
- Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje*.
- Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos*.
- Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases*.
- Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos*.
- Desarrolla programas, aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación*.
- Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información*.
- Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y la consistencia de los datos*.
- Comprende el origen de la orientación a objetos y sabe por qué se utilizan los objetos como clave para descomponer los sistemas en módulos en lugar de la funcionalidad.
- Conoce los principales conceptos de la tecnología de objetos y su aplicación en el diseño de software.
- Utiliza los principios de diseño de la orientación a objetos y aplicarlos en la resolución de problemas prácticos.
- Conoce las notaciones básicas de diseño que permiten representar las clases, sus relaciones y los objetos.
- Comprende el concepto de polimorfismo y de ligadura dinámica, y sabe aplicarlos en el diseño de clases.
- Conoce el concepto de herencia, sus distintas formas y los problemas originados en el diseño de lenguajes de programación.
- Comprende las relaciones complejas entre el sistema de tipos, la herencia y el polimorfismo.
- Posee habilidades para aplicar los conceptos de herencia, polimorfismo y ligadura dinámica para realizar diseños e implementaciones reutilizables.
- Sabe aplicar en la resolución de problemas concretos los principios de diseño con el objetivo de especificar jerarquías de clases y comprender los criterios para escoger entre una relación de herencia o de clientela.
- Caracteriza las instrucciones en lenguaje máquina y en lenguaje ensamblador. Distingue los diferentes formatos de las instrucciones y describe los diferentes métodos para codificar el código de operación.
- Distingue entre los diferentes tipos de instrucciones en ensamblador, modos de direccionamiento, registros, clases de arquitecturas a nivel de lenguaje máquina y tipos de operandos.
- Implementa un mismo programa en código ensamblador y en código de alto nivel.
- Explica cómo se implementan construcciones de los lenguajes de alto nivel en ensamblador y cómo se representan y almacenan en el computador datos y estructuras (vectores, matrices y registros).
- Describe una implementación elemental de camino de datos y unidad de control.
- Explica cómo la unidad de control de una CPU interpreta una instrucción a nivel máquina tanto en implementaciones cableadas como microprogramadas.
- Explica la estructura y el funcionamiento de la jerarquía de memoria en un computador y mostrar la necesidad de su presencia.

	<ul style="list-style-type: none"> • Describe el hardware para gestión de la jerarquía de memoria en un computador (memoria cache, memoria virtual y protección). Describe el funcionamiento de la gestión de memoria virtual. • Describe cómo configurar y diseñar memorias utilizando varios módulos. Explica cómo incrementar el ancho y número de palabras, junto con el diseño de memoria entrelazada. • Describe las diferentes organizaciones de la memoria caché, analizando las posibles estrategias de extracción, colocación, reemplazo y actualización. Explica cómo se diseñaría una cache, analizando los parámetros que afectan a las prestaciones. • Explica las diferentes técnicas de gestión de E/S. Describe controladores o interfaces de dispositivo. • Explica el concepto de bus, estructuras y tipos. Describir los diferentes tipos de transferencia, el arbitraje, la temporización y el direccionamiento. • Distingue entre arquitecturas CISC/RISC. • Depura código a bajo nivel y desensambla. • Reconoce la importancia de la abstracción y conoce los tipos de abstracciones que aparecen en programación: funcional, de datos, de iteradores y abstracción por generalización. • Sabe diferenciar entre la especificación, representación e implementación de un tipo de dato abstracto, conociendo los conceptos de Función de Abstracción e Invariante de la Representación. • Comprende cómo los conceptos de ocultamiento de información y encapsulamiento ayudan al desarrollo de tipos de datos más fiables. • Comprende los métodos de especificación: basados en una definición mediante axiomas o el método constructivo u operacional (basado en el uso de precondiciones y postcondiciones). • Es capaz de diseñar e implementar pequeñas aplicaciones para cada uno de los distintos tipos de datos que se imparten en la materia (listas, pilas, colas, colas con prioridad, conjuntos, diccionarios, árboles, tablas hash, grafos). • Adquiere la capacidad para comprender cómo el uso de distintos tipos de datos afecta a la eficiencia de los algoritmos que la usan. • Es capaz de implementar en lenguajes de alto nivel los tipos de datos propios de la materia así como otros definidos por el usuario. • Conoce las distintas representaciones e implementaciones de los tipos de datos que se imparten en la materia. • Es capaz de comparar implementaciones alternativas para un tipo de dato analizando los factores que influyen en la eficiencia y el uso de memoria. • Posee la capacidad de evaluar las necesidades de una aplicación específica, tomando decisiones justificadas sobre los tipos de datos y la representación más adecuada.
<p>Resultados de Aprendizaje de la Asignatura</p>	<ul style="list-style-type: none"> • Comprende la necesidad de un proceso de traducción de un lenguaje de alto nivel. • Conoce y distingue los conceptos de algoritmo y programa. • Conoce los tipos de datos primitivos y sus operaciones. • Distingue entre tipo de dato y objeto. • Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje*. • Reconoce la importancia de la abstracción y conoce los tipos de abstracciones que aparecen en programación: funcional, de datos, de iteradores y abstracción por generalización. • Sabe diferenciar entre la especificación, representación e implementación de un tipo de dato abstracto, conociendo los conceptos de Función de Abstracción e Invariante de la Representación.

- Comprende los métodos de especificación: basados en una definición mediante axiomas o el método constructivo u operacional (basado en el uso de precondiciones y postcondiciones).
- Es capaz de diseñar e implementar pequeñas aplicaciones para cada uno de los distintos tipos de datos que se imparten en la materia (listas, pilas, colas, colas con prioridad, conjuntos, diccionarios, árboles, tablas hash, grafos).
- Adquiere la capacidad para comprender cómo el uso de distintos tipos de datos afecta a la eficiencia de los algoritmos que la usan.
- Es capaz de implementar en lenguajes de alto nivel los tipos de datos propios de la materia así como otros definidos por el usuario.
- Conoce las distintas representaciones e implementaciones de los tipos de datos que se imparten en la materia.
- Es capaz de comparar implementaciones alternativas para un tipo de dato analizando los factores que influyen en la eficiencia y el uso de memoria.
- Posee la capacidad de evaluar las necesidades de una aplicación específica, tomando decisiones justificadas sobre los tipos de datos y la representación más adecuada.

Plan de Evaluación

En el sistema de evaluación de la Universidad Internacional Isabel I de Castilla, en coherencia con la consecución gradual de competencias y resultados de aprendizaje que se ha descrito en la metodología, se dará preferencia a la evaluación continua complementada con una evaluación final presencial en cada unidad trimestral. Estas evaluaciones finales presenciales permiten obtener garantías respecto a la identidad del estudiante a la que se refiere la Guía de Apoyo para la elaboración de la Memoria de verificación de títulos oficiales universitarios (Grado y máster¹) y a la veracidad del trabajo realizado durante el proceso de aprendizaje online, puesto que una parte importante de estas pruebas finales consiste en pruebas de verificación de la evaluación continua. Ésta será, por tanto, la vía preferente y recomendada por la Universidad para la obtención de los mejores resultados por parte del estudiante.

Sin embargo, es voluntad de esta Universidad ofrecer también una respuesta adecuada para aquellas personas que, por razones personales o profesionales, no pueden hacer un seguimiento de las asignaturas mediante el sistema de evaluación continua. No podemos olvidar que el perfil característico del estudiante de las universidades no presenciales se corresponde con personas de más de 25 años, en muchos casos con otros estudios universitarios y con responsabilidades profesionales y personales que deben compatibilizar con sus estudios online.

Teniendo en cuenta ambas perspectivas, el sistema de evaluación de la Universidad Internacional Isabel I de Castilla queda configurado de la siguiente manera:

Opción 1. Evaluación continua más evaluación final. Los estudiantes que opten por esta vía podrán obtener hasta el 60% de la nota final a través de las actividades que se planteen en la evaluación continua. El 40% restante se podrá obtener en la prueba de evaluación final que se realizará de manera presencial. Esta prueba tendrá una parte dedicada a la verificación del trabajo realizado por el estudiante durante la evaluación continua (que se corresponde con el 60% de la nota final) y otra parte en la que realizarán diferentes pruebas teórico-prácticas para evaluar las competencias previstas en cada asignatura. La no superación de la parte de verificación implica que la calificación de la evaluación continua no se tendrá en cuenta y, por tanto, el 100% de la nota dependerá del resultado obtenido en la prueba final de evaluación de competencias.

Opción 2. Evaluación final. Para los estudiantes que opten por esta vía, el 100% de la nota de la asignatura depende del resultado obtenido en esta prueba de evaluación final. Tanto en el proceso de información previa como en la formalización de la matrícula, el tutor informará de la existencia de esta posibilidad y valorará conjuntamente con cada persona su experiencia previa en la temática de la asignatura y otros factores que puedan influir en el resultado final.

Todos los estudiantes, independientemente de la opción seleccionada, tendrán derecho a una convocatoria extraordinaria de la prueba final de evaluación de competencias que se realizará un mes después de finalizadas las pruebas de evaluación final ordinaria del conjunto de tres trimestres. Para los estudiantes de evaluación continua que no hayan superado la verificación y que también hayan suspendido la prueba de evaluación de competencias ordinaria, el 100% de la nota final dependerá del resultado obtenido en esta convocatoria extraordinaria o "Prueba de conjunto".

¹ Versión 0.1 - 22/03/2011 (Disponible en: http://www.aneca.es/content/download/10717/120032/file/verifica_guia_11°324.pdf)

Opciones	Seguimiento de la Evaluación Continua (EC)	Ponderación valor%		Opciones	Examen final de <i>verificación</i> de la EC	Examen final de validación de competencias	Total
Opción 1.	Si	60%	→	Opción 1.	Superado.	40%	100%
					No superado.	100%	100%
Opción 2.	No	0%	→	Opción 2.	No.	100%	100%

Tabla. Sistema de evaluación.

Nota: Si no se supera la verificación se pasa de la Opción 1 de evaluación a la Opción 2.

Los alumnos que no superen alguno/s de los exámenes finales trimestrales de validación de competencias pasarán a la evaluación extraordinaria que se celebrará un mes después de cada conjunto de tres trimestres y que se denominará "Prueba de conjunto".

Finalmente, las Prácticas externas y el Trabajo Fin de Grado (TFG) tendrán su propio sistema de evaluación, que se especificará en las Guías docentes correspondientes. El TFG, en todo caso, deberá ser defendido por el estudiante ante una Comisión de Evaluación.

El sistema de evaluación final será común para todas las asignaturas de la materia y se basará en una selección de las pruebas de evaluación más adecuadas para el tipo de competencias que se trabajen. Las pruebas de evaluación, on-line o presenciales, se clasifican de la siguiente forma (Montanero et al., 2006):

1. Pruebas para evaluar competencias relacionadas con la comprensión, análisis, expresión de información (1, 2, 3, 4, 12).
2. Pruebas para evaluar competencias relacionadas con la aplicación de técnicas, procedimientos o protocolos de actuación y resolución de problemas (5, 6, 7, 13).
3. Pruebas para evaluar competencias relacionadas con la capacidad de investigar, pensar o actuar con creatividad y comunicarse verbalmente (8, 9, 12).
4. Pruebas para evaluar otras competencias profesionales, sociales y personales de carácter transversal (6, 9, 10, 11, 12).

² Montanero, M.; Mateos, V. L.; Gómez, V.; Alejo, R.: Orientaciones para la elaboración del Plan Docente de una Asignatura. Guía extensa. Badajoz, Universidad de Extremadura, Servicio de Publicaciones. 2006

Estrategias Evaluativas	Componentes de las competencias		
	Saber Competencias técnicas	Saber Hacer Competencias metodológicas	Saber ser-estar Competencias sociales y personales
Pruebas objetivas (tipo test).	x		
Pruebas semiobjetivas (preguntas cortas).	x		
Pruebas de desarrollo.	x		
Entrevista oral (en determinadas áreas).	x		x
Solución de problemas.	x	x	
Análisis de casos o supuestos prácticos.	x	x	x
Registros de observación sistemática.	x		
Proyectos y trabajos.	x	x	x
Entrevista (tutoría ECTS).	x	x	x
Pruebas de ejecución.	x	x	x
Solución de problemas.	x	x	x
Prueba de evaluación presencial.	x	x	x
Otros.			

Tabla. Estrategias o procedimientos de evaluación.

Los procedimientos de evaluación, al igual que ocurre con las actividades, se integran en el Sistema de Garantía Interna de Calidad (SGIC) de esta Universidad, de manera que la información recogida en cada trimestre se tendrá en cuenta en posteriores implementaciones de las asignaturas. La información acerca de la evaluación formará parte del compromiso público de la Universidad Internacional Isabel I de Castilla con sus estudiantes, de manera que las Guías docentes proporcionarán la información precisa sobre cómo se va a realizar el seguimiento de su trabajo y en qué va a consistir el sistema de evaluación de cada asignatura.

El sistema de calificaciones previsto para esta titulación se ajusta al Real Decreto 1125/2003, de 5 de septiembre, por el que se establece el sistema europeo de créditos y el sistema de calificaciones en las titulaciones universitarias de carácter oficial y de validez en todo el territorio nacional, que en su artículo 5, respecto al Sistema de calificaciones establece lo siguiente: La obtención de los créditos correspondientes a una materia comportará haber superado los exámenes o pruebas de evaluación correspondientes.

	<p>El nivel de aprendizaje conseguido por los estudiantes se expresará con calificaciones numéricas, que se reflejarán en su expediente académico junto con el porcentaje de distribución de estas calificaciones, sobre el total de alumnos que hayan cursado los estudios de la titulación en cada curso académico.</p> <p>La media del expediente académico de cada alumno será el resultado de la aplicación de la siguiente fórmula: suma de los créditos obtenidos por el alumno multiplicados cada uno de ellos por el valor de las calificaciones que correspondan, y dividida por el número de créditos totales obtenidos por el alumno.</p> <p>Los resultados obtenidos por el alumno en cada una de las materias del plan de estudios se calificarán en función de la siguiente escala numérica de 0 a 10, con expresión de un decimal, a la que podrá añadirse su correspondiente calificación cualitativa: 0-4,9: Suspenso (SS). 5,0-6,9: Aprobado (AP).7,0 -8,9: Notable (NT).9,0 -10: Sobresaliente (SB).</p> <p>Los créditos obtenidos por reconocimiento de créditos correspondientes a actividades formativas no integradas en el plan de estudios no serán calificados numéricamente ni computarán a efectos de cómputo de la media del expediente académico.</p> <p>La mención de Matrícula de Honor podrá ser otorgada a alumnos que hayan obtenido una calificación igual o superior a 9.0. Su número no podrá exceder del cinco por ciento de los alumnos matriculados en una materia en el correspondiente curso académico, salvo que el número de alumnos matriculados sea inferior a 20, en cuyo caso se podrá conceder una sola Matrícula de Honor.</p>
<p>Sistema de Calificación</p>	<p>Ponderación de la Evaluación Continua dentro del Proceso: 60%</p> <ul style="list-style-type: none"> • Estudio de Caso Real de aplicación práctica: 10% • Contenidos teóricos/Texto Canónico: 20% • Foros de Debate: 15% • Trabajo Colaborativo/WebQuest: 15% <p>Ponderación de la Evaluación Final dentro del Proceso: 40%</p> <ul style="list-style-type: none"> • Prueba de Contenidos + Prueba de Validación del Alumno/a

<p>Introducción</p>	<p>Esta asignatura introduce, tras de programación, el estudio de la eficiencia de algoritmos, así como los Tipos de Datos Abstractos (TDA), analizando la especificación e implementación de los más comunes: listas, pilas, colas, árboles, tablas hash o grafos.</p>
<p>Breve Descripción de los Contenidos</p>	<p>UNIDAD DIDÁCTICA 1 – Algorítmica. Introducción y Conceptos Básicos.</p> <ol style="list-style-type: none"> 1. Introducción a la Algorítmica <ol style="list-style-type: none"> a. Tipos de lenguajes de programación. b. Fases de la programación. 2. Representación de algoritmos. <ol style="list-style-type: none"> a. Datos. Definición y clasificación. b. Constantes y variables. c. Operadores y expresiones. d. Entrada y salida de información. e. Sistemas de representación de algoritmos. <p>UNIDAD DIDÁCTICA 2 – Complejidad de algoritmos.</p> <ol style="list-style-type: none"> 1. Complejidad de algoritmos. Costes y eficiencia. <ol style="list-style-type: none"> a. Tipos de complejidad. b. Análisis de las estructuras de control. c. Análisis de algoritmos combinados. d. Análisis de casos. 2. Notación asintótica. <ol style="list-style-type: none"> a. Tipología. b. Notación O. <p>UNIDAD DIDÁCTICA 3 – Estructuras secuenciales.</p> <ol style="list-style-type: none"> 1. Estudio de pilas. 2. Estudio de cola. 3. Estudio de listas. 4. Estructuras encadenadas. 5. Ejemplos de implementación <p>UNIDAD DIDÁCTICA 4 – Árboles.</p> <ol style="list-style-type: none"> 1. Árbol y árbol binario. Definición <ol style="list-style-type: none"> a. Aspectos generales. b. Árbol binario. Peculiaridades y características. 2. Tipología de recorridos.

3. Ejemplos de implementación.

UNIDAD DIDÁCTICA 5 – Grafos.

1. Grafo. Definición.
2. Propiedades de un grafo.
3. Algoritmos destacables.
4. Ejemplos de implementación.

UNIDAD DIDÁCTICA 6 – Tablas HASH.

1. Tablas Hash. Definición y concepto.
2. Propiedades de una tabla Hash.
3. Ejemplos de implementación.

<p>Bibliografía Básica</p>	<p>Dos libros con un resumen de no más de 10 líneas del comentario valorativo sobre el libro expuesto en los materiales. Indicación de los capítulos o , incluso, páginas recomendadas JOYANES AGUILAR, LUIS: <i>Fundamentos de programación. Algoritmos y estructuras de datos.</i> 2ª Edición. Ed. MacGraw-Hill. 1996.</p> <p>Libro fundamental para esta asignatura. Comprende, de forma amplia y exhaustiva, los contenidos a estudiar. Dispone de numerosos esquemas, diagramas y listados de código, tanto en lenguajes de programación como C o PASCAL. Aporta también ejemplos de algoritmos, implementados en pseudo-código. Finalmente, reseñar que dispone de un conjunto de anexo donde encontramos varias guías con la sintaxis correspondiente a varios lenguajes de programación, entre los cuales está el lenguaje de programación C, en concreto el ANSI C. WIRTH, NIKLAUS: <i>Algoritmos Mas Estructuras de Datos Igual a Programas.</i> Ediciones del Castillo. 1986.</p>
<p>Bibliografía Complementaria</p>	<p>Hernández, Z.J. y otros: Fundamentos de Estructuras de Datos. Soluciones en Ada, Java y C++, Thomson, 2005. Martí Ollet, N., Ortega Mallén, Y., Verdejo López, J.A.: <i>Estructura de datos y algoritmos. Ejercicios y problemas resueltos</i>, Pearson Prentice Hall, 2003. Campos Laclaustra, J.: <i>Estructuras de Datos y Algoritmos</i>, Prensas Universitarias de Zaragoza, Colección Textos Docentes, 1995. Franch Gutiérrez, X.: Estructuras de Datos. Especificación, Diseño e Implementación, 3ª edición, Ed. Edicions UPC, 2001. Mehta, D.P. y Sahni, S.: <i>Handbook of Data Structures and Applications</i>, Chapman & Hall/CRC, 2005. Thomas H. Cormen, Charles E. Leiserson, y Ronald L. Rivest: <i>Introduction to Algorithms</i>, 2nd Edition, McGraw-Hill, 2001. Robert Sedgwick: <i>Algorithms in C</i>, 3rd edition, Addison Wesley, 1998. Brian W. Kernighan y Dennis M. Ritchie: <i>El Lenguaje de Programación C</i>, Prentice Hall. 1978</p>
<p>Otros Recursos</p>	<p>http://www.algoritmia.net/ - Web sobre diseño de algoritmos, con numerosos foros donde encontrar respuestas a todo tipo de dudas sobre estos temas. http://www.conclase.net/ - Web sobre programación. Incluye sección para el lenguaje de programación C, con interesantes ejemplos y aplicaciones de las distintas estructuras de datos. http://www.brpreiss.com/books/opus4/ - Libro on-line. Programación de estructuras de datos y algoritmos en C++. http://www.lawebdelprogramador.com/ - Página web con multitud de recursos orientados a la programación. http://www.carlospes.com/curso_de_algoritmos/ - Interesante curso on-line sobre diseño de algoritmos. http://es.wikibooks.org/wiki/Estructuras_de_datos_y_algoritmos - Wiki libro sobre estructuras de datos y algoritmos, con enlaces sobre conceptos relevantes vinculados a este tema. http://xlinux.nist.gov/dads// - Web del NIST. Es un diccionario que alberga los principales términos sobre algoritmos y estructuras de datos. Completamente en inglés. http://www.programacion.com/articulo/estructuras_de_datos_y_algoritmos_en_java_309 - Curso sobre estructuras de datos y algoritmos con aplicación práctica sobre el lenguaje java. http://www.cs.sunysb.edu/~algorithm/ - Una colección de implementaciones de algoritmos para problemas fundamentales de algoritmos de combinatoria</p>

	<p>http://www.cs.princeton.edu/courses/archive/fall03/cs528/ - Curso de estructura de datos y algoritmos de la universidad de Princeton.</p> <p>http://www-cs-faculty.stanford.edu/~uno/sgb.html- Libro on-line con explicaciones actualizadas de muchos algoritmos y conjuntos de datos.</p> <p>http://www.acm.org/repository/ - Extenso repositorio de artículos sobre temática relacionada con esta asignatura.</p> <p>http://aofa.cs.purdue.edu/ - Página dedicada al análisis de los más variados tipos de algoritmos.</p> <p>http://courses.csail.mit.edu/6.854/current/ - Curso sobre algorítmica avanzada ofrecido por el MIT.</p> <p>https://sites.google.com/site/estdatjiq/home - Curso sobre estructuras de datos con múltiples recursos y apuntes</p>
--	--