



DATOS IDENTIFICATIVOS

Informática: Algoritmos y estructuras de datos I

Asignatura	Informática: Algoritmos y estructuras de datos I			
Código	O06G150V01201			
Titulación	Grado en Ingeniería Informática			
Descriptores	Creditos ECTS	Seleccione	Curso	Cuatrimestre
	6	FB	1	2c
Lengua	Castellano			
Impartición	Gallego			
Departamento				
Coordinador/a	Laza Fidalgo, Rosalia			
Profesorado	Fernández Riverola, Florentino Laza Fidalgo, Rosalia Pavón Rial, María Reyes			
Correo-e	rlaza@uvigo.es			
Web	http://fatic.uvigo.es			
Descripción general	Esta asignatura es obligatoria en el segundo semestre de primer curso. Es una continuación de la asignatura de programación impartida en el primer curso. Esta asignatura capacita al alumno para enfrentarse a problemas de programación complejos imprescindibles para cursar las siguientes asignaturas del plan de estudios. En esta asignatura no se emplea el inglés como lengua de impartición ni en el material docente.			

Competencias

Código	
A2	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.
B8	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.
C3	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
C4	Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería
C12	Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos
C13	Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema
C22	Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software
C25	Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software
C28	Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales
D1	I1: Capacidad de análisis, síntesis y evaluación
D5	I5: Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales
D6	I6: Capacidad de diseñar y realizar experimentos sencillos y analizar e interpretar sus resultados
D7	I7: Capacidad de buscar, relacionar y estructurar información proveniente de diversas fuentes y de integrar ideas y conocimientos
D8	I8: Resolución de problemas
D9	I9: Capacidad de tomar decisiones

D11	P1: Capacidad de actuar autónomamente
D15	P5: Capacidad de relación interpersonal
D16	S1: Razonamiento crítico
D18	S3: Aprendizaje autónomo
D20	S5: Creatividad
D22	S7: Tener iniciativa y ser resolutivo
D24	S9: Tener motivación por la calidad y la mejora continua

Resultados de aprendizaje

Resultados previstos en la materia	Resultados de Formación y Aprendizaje			
RA1: Concebir, desarrollar y utilizar de forma eficiente los tipos de datos y estructuras más adecuados a un problema.	A2	B8	C3 C4 C12 C13 C22 C28	D1 D5 D6 D7 D8 D9 D11 D15 D16 D18 D20 D22 D24
RA2: Encontrar soluciones algorítmicas a problemas, comprendiendo la idoneidad y complejidad de las soluciones propuestas.	A2	B8	C3 C12 C13 C25	D1 D5 D6 D7 D8 D9 D11 D15 D16 D18 D20 D22 D24
RA3: Determinar la complejidad en tiempo y espacio de diferentes algoritmos.	A2	B8	C3 C12 C13	D1 D5 D6 D8 D9 D11 D15 D16 D18 D20 D22 D24
RA4: Conocer la recursividad como herramienta de construcción de programas.	A2	B8	C12 C22 C25 C28	D1 D5 D6 D8 D9 D11 D15 D16 D18 D20 D22 D24

RA5: Programar aplicaciones de forma fuerte, correcta y eficiente teniendo en cuenta restricciones de tiempo y coste, y eligiendo el paradigma y los lenguajes de programación más adecuados.	A2	B8	C25 C28	D1 D5 D6 D7 D8 D9 D11 D15 D16 D18 D20 D22 D24
RA6: Conocer nuevas técnicas de programación, en particular el uso de memoria dinámica y las estructuras de datos enlazadas que están en la base de muchas aplicaciones.	A2	B8	C3 C12 C25 C28	D1 D5 D6 D8 D9 D11 D15 D16 D18 D20 D22 D24
RA7: Usar las herramientas de un entorno de desarrollo de programación para crear y desarrollar aplicaciones.	A2	B8	C3 C4 C13 C25 C28	D1 D5 D6 D8 D9 D11 D15 D16 D18 D20 D22 D24
RA8: Saber analizar, especificar e implementar estructuras de datos lineales desde una perspectiva de los TAD.	A2	B8	C13 C25	D1 D5 D6 D7 D8 D9 D11 D15 D16 D18 D20 D22 D24
RA9: Saber resolver problemas utilizando los TAD más apropiados.	A2	B8	C3 C12 C13 C22	D1 D5 D6 D7 D8 D9 D11 D15 D16 D18 D20 D22 D24

RA10: Conocer el funcionamiento y las técnicas básicas de ordenación de la información y la consulta eficiente de la misma.

A2 B8 C12 D1
C13 D5
C22 D6
C28 D8
D9
D11
D15
D16
D18
D20
D22
D24

Contenidos

Tema

Análisis de la eficiencia de algoritmos.	- Notaciones Asintóticas. - Análisis de algoritmos. - Reglas prácticas para el cálculo de eficiencia.
Estructuras de datos dinámicas.	- Las referencias como enlace. - Gestión de estructuras enlazadas. - Estructura enlazada simple. - Estructura doblemente enlazada. - Estructura circular - Nodo centinela - Gestión de estructuras enlazadas con nodos centinela
Tipos abstractos de datos. Estructuras lineales.	- Abstracción - TAD Pila - TAD Cola - TAD Lista
Algoritmos de búsqueda.	- Búsqueda Lineal. - Búsqueda Binaria. - Búsqueda Hashing.
Diseño de algoritmos recursivos.	- Ejemplos de recursividad. - Recursividad y variables locales.
Técnicas de Verificación y Pruebas	- Fundamentos de prueba del software - Pruebas de Caja Blanca - Pruebas de Caja Negra - Estrategias de prueba
Algoritmos de ordenación	- Ordenación por Inserción. - Ordenación por Selección. - Ordenación Burbuja. - Ordenación Shell. - Ordenación QuickSort. - Ordenación MergeSort

Planificación

	Horas en clase	Horas fuera de clase	Horas totales
Lección magistral	8	24	32
Aprendizaje basado en proyectos	4	8	12
Trabajo tutelado	7	21	28
Aprendizaje colaborativo.	0	1.975	1.975
Prácticas de laboratorio	20	40	60
Resolución de problemas y/o ejercicios	4.5	5.625	10.125
Proyecto	2	2.5	4.5
Presentación	0.7	0.7	1.4

*Los datos que aparecen en la tabla de planificación son de carácter orientativo, considerando la heterogeneidad de alumnado

Metodologías

	Descripción
Lección magistral	Consisten en clases magistrales donde se impartirá la base teórica de la materia y se expondrán ejemplos aclaratorios, además de establecer la relación existente entre los diferentes temas.

Aprendizaje basado en proyectos	Trabajo en grupo, en el que tienen que realizar un proyecto empleando el lenguaje de programación JAVA. Cada miembro del equipo tiene que implementar una parte del proyecto, para finalmente integrar todo obteniendo un producto final. Se realizará tanto en horas presenciales de laboratorio, como en horas no presenciales.
Trabajo tutelado	Trabajo colaborativo en el aula, una vez realizada la clase magistral y el alumno/a trabajado el tema en las horas no presenciales; se realizarán diferentes tipos de actividades en el aula para reforzar los conocimientos adquiridos.
Aprendizaje colaborativo.	Cada miembro del grupo del proyecto desarrollado, deberá explicar su parte a cada uno de sus compañeros. De forma que todos tengan un control absoluto de la totalidad del proyecto.
Prácticas de laboratorio	Aplicación a nivel práctico de la teoría de un ámbito de conocimiento en un contexto determinado. Ejercicios prácticos a través de los laboratorios. Se empleará para la resolución de problemas a lenguaje de programación JAVA.

Atención personalizada

Metodologías	Descripción
Trabajo tutelado	Tiempo reservado para atender y resolver dudas.
Prácticas de laboratorio	Tiempo reservado para atender y resolver dudas.
Aprendizaje colaborativo.	Tiempo reservado para atender y resolver dudas.

Evaluación

	Descripción	Calificación	Resultados de Formación y Aprendizaje			
Trabajo tutelado	Durante el trabajo colaborativo realizado en el aula. Se emplearán diferentes metodologías docentes para conseguir la participación activa del alumnado. Las distintas actividades serán evaluadas por el profesor. Resultados de aprendizaje: RA1, RA2, RA3, RA4, RA6, RA8, RA9, RA10	10	A2	B8	C3 C4 C12 C13 C22 C25 C28	D1 D5 D6 D7 D8 D9 D11 D15 D16 D18 D20 D22 D24
Resolución de problemas y/o ejercicios	Actividad en la que se formulan problemas y/o ejercicios relacionados con los algoritmos y estructuras de datos. El alumno debe desarrollar en Java las soluciones adecuadas y correctas de forma individual. Resultados de aprendizaje: RA1, RA2, RA3, RA4, RA6, RA8, RA9, RA10	60	A2	B8	C3 C4 C12 C13 C22 C25 C28	D1 D5 D6 D7 D8 D9 D11 D15 D16 D18 D20 D22 D24
Proyecto	El final del cuatrimestre, el alumno realizará una prueba individual, en la que tendrá que modificar el proyecto realizado durante lo cuatrimestre. Resultados de aprendizaje: RA1, RA5, RA7, RA9, RA10	20	A2	B8	C3 C4 C12 C13 C22 C25 C28	D1 D5 D6 D7 D8 D9 D11 D15 D16 D18 D20 D22 D24

Presentación	Al final del cuatrimestre, el profesor realizará una evaluación conjunta sobre el proyecto desarrollado. El objetivo es comprobar que todos los miembros del grupo entienden la totalidad del proyecto.	10	A2	B8	C3	D1
					C4	D5
					C12	D6
					C13	D7
	Resultados de aprendizaje: RA1, RA5, RA7, RA9, RA10				C22	D8
					C25	D9
					C28	D11
						D15
						D16
						D18
						D20
						D22
						D24

Otros comentarios sobre la Evaluación

CRITERIOS DE EVALUACIÓN PARA ASISTENTE 1 EDICIÓN DE ACTAS

Calificación final = $0.1 * (\text{nota trabajos de aula}) + 0.1 * (\text{nota de la presentación en grupo sobre el proyecto}) + 0.2 * (\text{nota del proyecto}) + 0.6 * (\text{nota media de las tres pruebas de resoluciones de problemas})$

El primer punto se obtiene ponerlo trabajo colaborativo realizado en horario presencial, el siguiente punto por las exposiciones realizadas sobre el proyecto, dos puntos máximo por la evaluación individual del proyecto y seis puntos máximo por la evaluación individual de resolución de problemas.

Para poder aplicar la fórmula anterior es necesario que el alumno obtenga como mínimo un 4 en la nota del proyecto y en la media de las pruebas de resoluciones de problemas.

Todos los estudiantes que se presenten a cualquiera de las pruebas se entenderá que siguen la materia de forma presencial y por lo tanto deberán seguir el procedimiento de evaluación descrito anteriormente. Si un estudiante no se presenta a alguna de las pruebas se le asignará una calificación de 0 en ellas.

Proceso de calificación de las actas

Para superar la asignatura a calificación final debe ser igual o superior a 5. En el caso de suspender, se guardará para la 2a. convocatoria cualquiera de las partes aprobadas (proyecto o resolución de problemas), y se calificará en el acta con la nota media de las dos partes, en caso de superar el 5 en esa media se pondrá una calificación de 4.

CRITERIOS DE EVALUACIÓN PARA NO ASISTENTE 1 EDICIÓN DE ACTAS

Calificación final = $0.3 * (\text{nota del proyecto}) + 0.7 * (\text{nota media de las tres pruebas de resoluciones de problemas})$

ES necesario que el alumno obtenga como mínimo un 4 en la nota del proyecto y en la media de las pruebas de resoluciones de problemas.

Las competencias y los resultados de aprendizaje son los mismos que para los asistente, porque hacen exactamente el mismo trabajo; la única diferencia es la resolución de ejercicios realizados en el aula, y la exposición del proyecto.

Proceso de calificación de las actas

Para superar la asignatura a calificación final debe ser igual o superior a 5. En el caso de suspender, se guardará para la 2la. convocatoria cualquiera de las partes aprobadas (proyecto o teoría), y se calificará en el acta con la nota media de las dos partes, en caso de superar el 5 en esa media se pondrá una calificación de 4.

CRITERIOS DE EVALUACIÓN PARA 2 EDICIÓN DE ACTAS

Calificación final = $0.3 * (\text{nota del proyecto}) + 0.7 * (\text{nota de resoluciones de problemas y ejercicios})$

Las competencias y los resultados de aprendizaje son los mismos que para los asistente, porque hacen exactamente el mismo trabajo; la única diferencia es la resolución de ejercicios realizados en el aula, y la exposición del proyecto.

Proceso de calificación de las actas

Para superar la asignatura a calificación final debe ser igual o superior a 5. ES necesario que el alumno obtenga como mínimo un 4, en la nota del proyecto y en la nota de resoluciones de problemas y ejercicios.

CRITERIOS DE EVALUACIÓN PARA NO ASISTENTE 2 EDICIÓN DE ACTAS

Calificación final = $0.3 * (\text{nota del proyecto}) + 0.7 * (\text{nota media de las tres pruebas de resoluciones de problemas})$

ES necesario que el alumno obtenga como mínimo un 4 en la nota del proyecto y en la media de las pruebas de resoluciones de problemas.

Las competencias y los resultados de aprendizaje son los mismos que para los asistentes, porque hacen exactamente el mismo trabajo; la única diferencia es la resolución de ejercicios realizados en el aula, y la exposición del proyecto.

Proceso de calificación de las actas

Para superar la asignatura a calificación final debe ser igual o superior a 5. ES necesario que el alumno obtenga como mínimo un 4, en la nota del proyecto y en la nota de resoluciones de problemas y ejercicios.

CRITERIOS DE EVALUACIÓN PARA FIN DE CARRERA PARA ASISTENTES Y NO ASISTENTES

Calificación final = nota de resoluciones de problemas y ejercicios

Proceso de calificación de las actas

Para superar la asignatura a calificación final debe ser igual o superior a 5.

FECHAS DE EVALUACIÓN:

El calendario de pruebas de evaluación aprobado oficialmente por la Junta de Centro de la ESEI que se encontrará publicado en la web

Fuentes de información

Bibliografía Básica

Brassard G., **Fundamentos de Algoritmia**, 4ª, Prentice Hall,

Laza R., **Metodología y Tecnología de la Programación**, 1ª, Pearson Prentice Hall,

Main M., **Data Structures and Other Objects Using Java**, 4ª, Pearson International Edition,

Goodrich M., Tamassia R., **Data structures and algorithms in Java**, 6ª, John Wiley & Sons,

Bibliografía Complementaria

Weiss, Mark Allen, **Data Structures and Algorithm Analysis in Java**, 3ª, Pearson,

Drozdek A., **Estructuras De Datos Y Algoritmos En Java**, 2ª, Thomson,

Joyanes L., Zahonero I., **Estructura de datos en Java**, McGrawHill,

Lewis J., Chase J., **Estructuras de datos con Java. Diseño de estructuras y algoritmos**, 2ª, Pearson Addison Wesley,

Lee R.C.T, Tseng S.S, Chang R.C., Tsai Y.T., **Introducción al diseño y análisis de algoritmos**, McGrawHill,

Weiss, Mark Allen, **Data Structures & Problem Solving Using Java**, 4ª, Pearson,

Pressman Roger S., **Ingeniería del software: un enfoque práctico**, McGrawHill,

Recomendaciones

Asignaturas que continúan el temario

Algoritmos y estructuras de datos II/O06G150V01302

Asignaturas que se recomienda cursar simultáneamente

Programación II/O06G150V01205

Asignaturas que se recomienda haber cursado previamente

Informática: Programación I/O06G150V01104

Otros comentarios

Los alumnos que tienen la asignatura convalidada se recomienda que asistan a clase, porque esta asignatura tiene su continuación en segundo curso. Muchos de los conceptos explicados en primer curso son necesarios para cursar a asignatura de segundo, y ya se dan por adquiridos.

No está permitido el uso de dispositivos móviles durante las clases de teoría y práctica.

"Se recuerda a todo el alumnado la prohibición del uso de dispositivos móviles o electrónicos y ordenadores portátiles durante actividades evaluables, en cumplimiento del artículo 13.2.d) del Estatuto del Estudiante Universitario, relativo a los deberes del estudiante universitario, que establece el deber de abstenerse de la utilización o la cooperación en procedimientos fraudulentos en las pruebas

de evaluación, en los trabajos que se realicen o en documentos oficiales de la universidad"
