



Guía docente				
Datos Identificativos				2021/22
Asignatura (*)	Teoría de la computación	Código	614G01039	
Titulación	Grao en Enxeñaría Informática			
Descritores				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	2º cuatrimestre	Tercero	Optativa	6
Idioma	CastellanoGallego			
Modalidad docente	Presencial			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinador/a	Graña Gil, Jorge	Correo electrónico	jorge.grana@udc.es	
Profesorado	Graña Gil, Jorge	Correo electrónico	jorge.grana@udc.es	
	Novo Bujan, Jorge		j.novo@udc.es	
	Rivas Villar, David		david.rivas.villar@udc.es	
Web	moodle.udc.es			
Descripción general	Se trata de una asignatura en la que destaca el carácter integrador de su contenido, ya que sirve de puente entre lo que podemos denominar una "visión de usuario" de los lenguajes informáticos, representada por la programación estándar, y una "visión generativa" de éstos, en la que el alumno construye y adecúa un lenguaje de programación en atención a sus requerimientos. Finalmente, se transmite también al alumno una visión formal de los fundamentos propios de la ciencia de la computación.			
Plan de contingencia	1. Modificaciones en los contenidos: Ninguna.  2. Metodologías: *Metodologías docentes que se mantienen: Todas.  *Metodologías docentes que se modifican: En caso de necesidad, todas las metodologías empleadas podrían aplicarse de modo no presencial.  3. Mecanismos de atención personalizada al alumnado: Atención continuada en Teams, Moodle y correo electrónico.  4. Modificaciones en la evaluación: No son necesarias.  *Observaciones de evaluación: Ninguna.  5. Modificaciones de la bibliografía o webgrafía: Ninguna.			

Competencias / Resultados del título	
Código	Competencias / Resultados del título
A39	Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.
A40	Capacidad para conocer los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico, sintáctico y semántico asociadas, y saber aplicarlas para la creación, diseño y procesamiento de lenguajes.



A41	Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.
B1	Capacidad de resolución de problemas
B2	Trabajo en equipo
B3	Capacidad de análisis y síntesis
B6	Toma de decisiones
B8	Capacidad de trabajar en un equipo interdisciplinar
C6	Valorar críticamente el conocimiento, la tecnología y la información disponible para resolver los problemas con los que deben enfrentarse.
C7	Asumir como profesional y ciudadano la importancia del aprendizaje a lo largo de la vida.

Resultados de aprendizaje			
Resultados de aprendizaje	Competencias / Resultados del título		
Conocer en profundidad la estructura y función de los sistemas de descripción y reconocimiento de lenguajes formales.	A39 A40	B6	C7
Estudiar los conceptos, modelos y técnicas relacionados con estas cuestiones.	A39 A40	B6	C7
Conocer las estructuras de datos y los algoritmos utilizados para implementar los distintos modelos de reconocimiento de lenguajes formales, así como sus posibles dominios de aplicación práctica.	A41	B6	C6 C7
Realizar implementaciones de estos modelos en alguno de esos dominios.	A41	B1 B2 B3	C6
Sintetizar todos los conceptos estudiados en ideas concretas que permitan comprender mejor los fundamentos de la computación.	A39	B6	C7
Perfeccionar las habilidades para realizar futuros trabajos de análisis, diseño y programación.	A40 A41	B1 B2 B3	C6
Considerar la integración de las técnicas y estructuras estudiadas aquí en otros dominios de aplicación.	A40 A41	B1 B2 B3 B8	C6

Contenidos	
Tema	Subtema
Preliminares sobre lenguajes formales	Alfabetos, palabras y lenguajes Lenguajes regulares y expresiones regulares Autómatas finitos
Lenguajes independientes del contexto y autómatas de pila	Gramáticas regulares Gramáticas regulares y lenguajes regulares Gramáticas independientes del contexto Arboles de derivación y ambigüedad Simplificación de gramáticas independientes del contexto Propiedades de los lenguajes independientes del contexto Algoritmos de análisis sintáctico Autómatas de pila Forma normal de Greibach



Máquinas de Turing	Definiciones básicas Máquinas de Turing como aceptadoras de lenguajes Construcción de máquinas de Turing Modificaciones de las máquinas de Turing Máquina de Turing universal
Lenguajes recursivamente enumerables	Lenguajes aceptados por máquinas de Turing Lenguajes regulares e independientes del contexto como lenguajes recursivos Propiedades de los lenguajes recursivos y recursivamente enumerables Gramáticas no restringidas y lenguajes recursivamente enumerables Lenguajes sensibles al contexto y la jerarquía de Chomsky
Resolubilidad	El problema de la parada El problema de correspondencia de Post Problemas no decidibles en lenguajes independientes del contexto
Computabilidad	Fundamentos de la teoría de funciones recursivas Alcance de las funciones recursivas primitivas Funciones recursivas parciales El poder de los lenguajes de programación

Planificación				
Metodologías / pruebas	Competencias / Resultados	Horas lectivas (presenciales y virtuales)	Horas trabajo autónomo	Horas totales
Sesión magistral	A39 A40 B8 C6 C7	18	36	54
Prácticas de laboratorio	A40 A41 B1 B2 B3 B6 B8 C6	13	26	39
Prueba de respuesta breve	A39 A40 B1 C6 C7	3	6	9
Solución de problemas	B1 B3 B6	4	20.5	24.5
Prueba objetiva	A39 A40 B1 C6 C7	3	16	19
Atención personalizada		4.5	0	4.5

(\*) Los datos que aparecen en la tabla de planificación són de carácter orientativo, considerando la heterogeneidad de los alumnos

Metodologías	
Metodologías	Descripción
Sesión magistral	La técnica que mejor se adapta a la impartición de los contenidos teóricos de esta asignatura está constituida por las clases magistrales. En ellas, haremos un uso intensivo de la pizarra virtual y de las transparencias, de modo que el ritmo de exposición de conceptos por parte del profesor y el de asimilación de los mismos por parte del alumno sean lo más acordes posible.
Prácticas de laboratorio	Estas prácticas serán utilizadas para implementar en algún lenguaje de programación los algoritmos más destacados, de entre todos aquéllos que hayan sido presentados en las sesiones teóricas.
Prueba de respuesta breve	Se realizarán controles al final de cada bloque temático, que permitirán al profesor conocer el grado de asimilación de la materia por parte de los alumnos, y modificar la estrategia docente si es necesario.
Solución de problemas	Se pondrán a disposición de los alumnos una serie de boletines de ejercicios, correspondientes a los bloques temáticos del programa de la asignatura. Los alumnos deberán entregar al profesor sus soluciones personales a estos ejercicios. El profesor deberá corregirlas, evaluarlas y comentarlas durante al menos una sesión.
Prueba objetiva	Se implementará bajo la forma de un examen final escrito.

Atención personalizada	
Metodologías	Descripción



Prácticas de laboratorio	<p>Dado el carácter personalizado de las prácticas de laboratorio y de las tutorías, estas actividades no deben dedicarse a extender los contenidos con nuevos conceptos, sino a aclarar los conceptos ya expuestos.</p> <p>El profesor debe además utilizarlas como una interacción que le permita extraer conclusiones respecto al grado de asimilación de la materia por parte de los alumnos.</p> <p>De esta manera, podrá desarrollar las clases magistrales y el resto de actividades no personalizadas atendiendo al progreso de los alumnos en las capacidades de comprensión y asimilación de los contenidos impartidos, compaginando el avance general de la clase con una atención específica a aquellos alumnos que presenten mayores dificultades en la tarea del aprendizaje y con un apoyo adicional a aquellos otros que presenten mayor soltura y deseen ampliar conocimientos.</p>
--------------------------	--

Evaluación			
Metodologías	Competencias / Resultados	Descripción	Calificación
Prueba objetiva	A39 A40 B1 C6 C7	Examen final escrito. (***)	0
Solución de problemas	B1 B3 B6	Boletines de ejercicios y controles de los mismos.	10
Prueba de respuesta breve	A39 A40 B1 C6 C7	Controles tipo test y cuestiones. (**)	60
Prácticas de laboratorio	A40 A41 B1 B2 B3 B6 B8 C6	Implementación de algoritmos en algún lenguaje de programación y resolución de problemas. (*)	30
Otros			

Observaciones evaluación
<p>(*) En las prácticas de laboratorio, se requiere que el alumno obtenga una nota mínima de 3 puntos (sobre 10).</p> <p>(**) La materia se dividirá en tres bloques temáticos. Al final de cada bloque temático, se realizará un control con cuestiones teóricas y prácticas. Cada control podrá consolidar hasta un 20% de la calificación. El porcentaje correspondiente a los controles no superados pasará a computarse en la prueba objetiva (examen final). Los alumnos que superen los tres controles, no tendrán que realizar el examen final.</p> <p>(***) En el caso de tener que realizar el examen final, se requiere que el alumno obtenga una nota mínima de 3 puntos (sobre 10). Los alumnos a tiempo parcial tendrán consideraciones adecuadas a su situación.</p>

Fuentes de información	
<b>Básica</b>	<ul style="list-style-type: none"> <li>- Thomas A. Sudkamp (1988). Languages and machines: an introduction to the theory of computer science. Addison Wesley</li> <li>- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2002). Introducción a la teoría de autómatas, lenguajes y computación. Addison Wesley</li> <li>- Dean Kelley (1995). Teoría de autómatas y lenguajes formales. Prentice Hall</li> </ul>
<b>Complementaria</b>	<ul style="list-style-type: none"> <li>- J. Glenn Brookshear (1993). Teoría de la computación: lenguajes formales, autómatas y complejidad. Addison Wesley Iberoamericana</li> <li>- Peter J. Denning, Jack B. Dennis, Joseph E. Qualitz (1978). Machines, languages and computation. Prentice Hall</li> <li>- Harry R. Lewis, Christos H. Papadimitriou (1998). Elements of the theory of computation. Prentice Hall</li> </ul>

Recomendaciones
Asignaturas que se recomienda haber cursado previamente



Programación I/614G01001

Matemática Discreta/614G01004

Programación II/614G01006

Álgebra/614G01010

Algoritmos/614G01011

Paradigmas de Programación/614G01014

**Asignaturas que se recomienda cursar simultáneamente**

**Asignaturas que continúan el temario**

Representación del Conocimiento y Razonamiento Automático/614G01036

Recuperación de la Información/614G01040

Diseño de los Lenguajes de Programación/614G01065

Procesamiento de Lenguajes/614G01067

**Otros comentarios**

(\*) La Guía Docente es el documento donde se visualiza la propuesta académica de la UDC. Este documento es público y no se puede modificar, salvo cosas excepcionales bajo la revisión del órgano competente de acuerdo a la normativa vigente que establece el proceso de elaboración de guías