



Guía docente				
Datos Identificativos				2015/16
Asignatura (*)	Teoría de Autómatas y Lenguajes Formal		Código	614111301
Titulación	Enxeñeiro en Informática			
Descritores				
Ciclo	Periodo	Curso	Tipo	Créditos
1º y 2º Ciclo	1º cuatrimestre	Tercero	Troncal	7.5
Idioma	CastellanoGallego			
Modalidad docente	Presencial			
Prerrequisitos				
Departamento	Computación			
Coordinador/a	Blanco Ferro, Antonio angel	Correo electrónico	antonio.blanco.ferro@udc.es	
Profesorado	Blanco Ferro, Antonio angel	Correo electrónico	antonio.blanco.ferro@udc.es	
Web	<a href="http://www.dc.fi.udc.es/~grana/TALF/">http://www.dc.fi.udc.es/~grana/TALF/</a>			
Descripción general	Se trata de una asignatura troncal, que se imparte de manera cuatrimestral en el tercer curso de la titulación de Ingeniero en Informática. 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.			

Competencias del título	
Código	Competencias del título
A1	Aprender de manera autónoma nuevos conocimientos y técnicas avanzadas adecuadas para la investigación, el diseño y el desarrollo de sistemas y servicios informáticos.
A3	Concebir y planificar el desarrollo de aplicaciones informáticas complejas o con requisitos especiales.
B1	Aprender a aprender.
B2	Resolver problemas de forma efectiva.
B3	Aplicar un pensamiento crítico, lógico y creativo.
B4	Aprendizaje autónomo.
B5	Trabajar de forma colaborativa.
B8	Trabajar en equipos de carácter interdisciplinar.
C6	Valorar críticamente el conocimiento, la tecnología y la información disponible para resolver los problemas con los que deben enfrentarse.

Resultados de aprendizaje			
Resultados de aprendizaje	Competencias del título		
Conocer en profundidad la estructura y función de los sistemas de descripción y reconocimiento de lenguajes formales.	A1	B1 B4	
Estudiar los conceptos, modelos y técnicas relacionados con estas cuestiones.	A1	B1 B4	
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.	A1 A3	B1 B4	C6
Realizar implementaciones de estos modelos en alguno de esos dominios.	A1 A3	B2 B3 B5	C6
Sintetizar todos los conceptos estudiados en ideas concretas que permitan comprender mejor los fundamentos de la computación	A1	B1 B4	



Perfeccionar las habilidades para realizar futuros trabajos de análisis, diseño y programación.	A1 A3	B2 B3 B5	C6
Considerar la integración de las técnicas y estructuras estudiadas aquí en otros dominios de aplicación.	A1 A3	B2 B3 B5 B8	C6

Contenidos	
Tema	Subtema
Preliminares matemáticos	Lógica elemental Teoría de conjuntos Relaciones y funciones Inducción matemática Cardinalidad
Lenguajes formales	Alfabetos, palabras y lenguajes Operaciones con palabras Operaciones con lenguajes
Lenguajes regulares y autómatas finitos	Lenguajes sobre alfabetos Lenguajes regulares y expresiones regulares Autómata finito determinista (AFD) Autómata finito no determinista (AFN) Equivalencia entre AFNs y AFDs Autómata finito con epsilon transiciones Autómatas finitos y expresiones regulares Aplicaciones prácticas de las expresiones regulares y de los autómatas finitos
Lenguajes independientes del contexto y autómatas de pila	Gramáticas regulares Gramáticas regulares y lenguajes regulares Gramáticas independientes del contexto Árboles 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
Introducción a la teoría de la complejidad computacional	Complejidad algorítmica Modelo general de cómputo y complejidad computacional Tiempo y espacio en máquinas de Turing Las distintas clases de complejidad Los problemas tratables y no tratables Reducibilidad en tiempo polinómico Problemas NP-completos

Planificación				
Metodologías / pruebas	Competencias	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Sesión magistral		30	60	90
Prácticas de laboratorio		10	20	30
Prueba de respuesta múltiple		4	4	8
Trabajos tutelados		1	5	6
Seminario		3	0	3
Aprendizaje colaborativo		4	4	8
Solución de problemas		3	16	19
Prueba de ensayo/desarrollo		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 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	Las prácticas de laboratorio tendrán horas de laboratorio reservadas, con ordenadores a disposición de los alumnos. Estas horas 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 múltiple	Se realizarán controles tipo test 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.
Trabajos tutelados	Los trabajos de grupos autónomos tutelados se realizarán a lo largo de todo el cuatrimestre. El profesor elegirá un tema de trabajo que será asignado por igual a todos los grupos. El tema será presentado por el profesor en una sesión en el aula, será desarrollado por los alumnos en horas no presenciales, y será supervisado y evaluado por el profesor en las tutorías en grupo. La evaluación se realizará a partir de la exposición de una memoria final por parte de los alumnos.
Seminario	Los seminarios se implementarán bajo la forma de un ciclo de charlas cortas o conferencias, sobre aplicaciones prácticas relacionadas con la materia de la asignatura. El objetivo de estas charlas es el de completar la percepción general que el alumno tiene sobre cómo los conceptos vistos en clase son puestos en práctica en la vida real.



Aprendizaje colaborativo	Los trabajos de grupos cooperativos se realizarán utilizando la "técnica puzzle" de Aronson. Según esta técnica, el profesor elige previamente un tema de trabajo y divide a los alumnos en varios grupos. A cada miembro de cada grupo se le asigna una sección de dicho tema, para que la estudie y la comente con los miembros de los otros grupos a los que les ha sido asignada la misma sección. Posteriormente, cada alumno regresa a su grupo y explica al resto de miembros su sección. Finalmente, el profesor realiza a los alumnos un test general sobre el tema elegido. Dicho test permitirá al profesor no sólo evaluar el grado de comprensión de los nuevos conocimientos adquiridos, sino también el nivel de cooperación que ha tenido lugar entre los miembros de cada grupo concreto.
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 en el aula.
Prueba de ensayo/desarrollo	Se implementará bajo la forma de un examen final escrito.

### Atención personalizada

Metodologías	Descripción
Prácticas de laboratorio Trabajos tutelados	<p>Dado el carácter personalizado de las prácticas de laboratorio, de los trabajos tutelados 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 desenvoltura y deseen ampliar conocimientos.</p>

### Evaluación

Metodologías	Competencias	Descripción	Calificación
Prácticas de laboratorio		Implementación de algoritmos en algún lenguaje de programación	25
Prueba de respuesta múltiple		Controles tipo test	7.5
Trabajos tutelados		Trabajo de grupos autónomos tutelados	5
Aprendizaje colaborativo		Trabajo de grupos cooperativos	7.5
Solución de problemas		Boletines de ejercicios	5
Prueba de ensayo/desarrollo		Examen final escrito	50
Otros			

### Observaciones evaluación

En el examen final se requiere una nota mínima de 3 puntos (sobre 10).

### Fuentes de información



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

## Recomendaciones

### Asignaturas que se recomienda haber cursado previamente

Inteligencia Artificial/614111404  
Compiladores/614111405  
Lenguajes Naturales/614111625

### Asignaturas que se recomienda cursar simultáneamente

### Asignaturas que continúan el temario

Estructura de Datos y de la Información/614111102  
Álgebra/614111106  
Matemática Discreta I/614111107  
Programación/614111109  
Algoritmos/614111206  
Programación Declarativa/614111207

### 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