



Guía Docente				
Datos Identificativos				2012/13
Asignatura (*)	Teoría da computación	Código	614G01039	
Titulación				
Descritores				
Ciclo	Período	Curso	Tipo	Créditos
Grao	2º cuatrimestre	Terceiro	Obrigatoria	6
Idioma	CastelánGalego			
Prerrequisitos				
Departamento	Computación			
Coordinación	Blanco Ferro, Antonio angel	Correo electrónico	antonio.blanco.ferro@udc.es	
Profesorado	Blanco Ferro, Antonio angel	Correo electrónico	antonio.blanco.ferro@udc.es	
Web				
Descrición xeral	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 &quot;visión de usuario&quot; de los lenguajes informáticos, representada por la programación estándar, y una &quot;visión generativa&quot; 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 da titulación	
Código	Competencias da titulación

Resultados da aprendizaxe			
Competencias de materia (Resultados de aprendizaxe)	Competencias da titulación		
Conocer en profundidad la estructura y función de los sistemas de descripción y reconocimiento de lenguajes formales.	A39 A40	B7 B14	
Estudiar los conceptos, modelos y técnicas relacionados con estas cuestiones.	A39 A40	B7 B14	
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	B7 B14	C6
Realizar implementaciones de estos modelos en alguno de esos dominios.	A41	B9 B10 B11	C6
Sintetizar todos los conceptos estudiados en ideas concretas que permitan comprender mejor los fundamentos de la computación	A39	B7 B14	
Perfeccionar las habilidades para realizar futuros trabajos de análisis, diseño y programación.	A40 A41	B9 B10 B11	C6
Considerar la integración de las técnicas y estructuras estudiadas aquí en otros dominios de aplicación.	A40 A41	B9 B10 B11 B16	C6

Contidos	
Temas	Subtemas
Preliminares sobre lenguajes formales	Alfabetos, palabras y lenguajes Lenguajes regulares y expresiones regulares Autómatas finitos



Lenguajes independientes del contexto y autómatas de pila	<p>Gramáticas regulares</p> <p>Gramáticas regulares y lenguajes regulares</p> <p>Gramáticas independientes del contexto</p> <p>Arboles de derivación y ambigüedad</p> <p>Simplificación de gramáticas independientes del contexto</p> <p>Propiedades de los lenguajes independientes del contexto</p> <p>Algoritmos de análisis sintáctico</p> <p>Autómatas de pila</p> <p>Forma normal de Greibach</p>
Máquinas de Turing	<p>Definiciones básicas</p> <p>Máquinas de Turing como aceptadoras de lenguajes</p> <p>Construcción de máquinas de Turing</p> <p>Modificaciones de las máquinas de Turing</p> <p>Máquina de Turing universal</p>
Lenguajes recursivamente enumerables	<p>Lenguajes aceptados por máquinas de Turing</p> <p>Lenguajes regulares e independientes del contexto como lenguajes recursivos</p> <p>Propiedades de los lenguajes recursivos y recursivamente enumerables</p> <p>Gramáticas no restringidas y lenguajes recursivamente enumerables</p> <p>Lenguajes sensibles al contexto y la jerarquía de Chomsky</p>
Resolubilidad	<p>El problema de la parada</p> <p>El problema de correspondencia de Post</p> <p>Problemas no decidibles en lenguajes independientes del contexto</p>
Computabilidad	<p>Fundamentos de la teoría de funciones recursivas</p> <p>Alcance de las funciones recursivas primitivas</p> <p>Funciones recursivas parciales</p> <p>El poder de los lenguajes de programación</p>
Introducción a la teoría de la complejidad computacional	<p>Complejidad algorítmica</p> <p>Modelo general de cómputo y complejidad computacional</p> <p>Tiempo y espacio en máquinas de Turing</p> <p>Las distintas clases de complejidad</p> <p>Los problemas tratables y no tratables</p> <p>Reducibilidad en tiempo polinómico</p> <p>Problemas NP-completos</p>

Planificación			
Metodoloxías / probas	Horas presenciais	Horas non presenciais / traballo autónomo	Horas totais
Sesión maxistral	18	36	54
Prácticas de laboratorio	13	26	39
Proba de resposta múltiple	2	4	6
Traballos tutelados	1	5.5	6.5
Solución de problemas	4	17	21
Proba de ensaio	3	16	19
Atención personalizada	4.5	0	4.5

\*Os datos que aparecen na táboa de planificación son de carácter orientativo, considerando a heteroxeneidade do alumnado

Metodoloxías	
Metodoloxías	Descrición



Sesión maxistral	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.
Proba de resposta 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.
Traballos 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.
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.
Proba de ensaio	Se implementará bajo la forma de un examen final escrito.

### Atención personalizada

Metodoloxías	Descrición
Prácticas de laboratorio Traballos 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>

### Avaliación

Metodoloxías	Descrición	Cualificación
Prácticas de laboratorio	Implementación de algoritmos en algún lenguaje de programación	25
Traballos tutelados	Trabajo de grupos autónomos tutelados	5
Proba de resposta múltiple	Controles tipo test	5
Solución de problemas	Boletines de ejercicios	5
Proba de ensaio	Examen final escrito	60
Outros		

### Observacións avaliación

&lt;p&gt;En el examen final se requiere una nota mínima de 3 puntos (sobre 10).&lt;/p&gt;
---



## Fontes de información

<b>Bibliografía 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>Bibliografía 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>

## Recomendacións

### Materias que se recomenda ter cursado previamente

Representación do Coñecemento e Razoamento Automático/614G01036

Recuperación da Información/614G01040

Deseño das Linguaxes de Prgramación/614G01065

Procesamento de Linguaxes/614G01067

### Materias que se recomenda cursar simultaneamente

### Materias que continúan o temario

Programación I/614G01001

Matemática Discreta/614G01004

Programación II/614G01006

Álgebra/614G01010

Algoritmos/614G01011

Paradigmas de Programación/614G01014

### Observacións

(\*A Guía docente é o documento onde se visualiza a proposta académica da UDC. Este documento é público e non se pode modificar, salvo casos excepcionais baixo a revisión do órgano competente dacordo coa normativa vixente que establece o proceso de elaboración de guías