



Guía Docente				
Datos Identificativos				2020/21
Asignatura (*)	Teoría da computación	Código	614G01039	
Titulación				
Descritores				
Ciclo	Período	Curso	Tipo	Créditos
Grao	2º cuatrimestre	Terceiro	Optativa	6
Idioma	CastelánGalego			
Modalidade docente	Híbrida			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinación	Graña Gil, Jorge	Correo electrónico	jorge.grana@udc.es	
Profesorado	Graña Gil, Jorge Novo Bujan, Jorge	Correo electrónico	jorge.grana@udc.es j.novo@udc.es	
Web	moodle.udc.es			
Descrición xeral	Trátase dunha materia na que destaca o carácter integrador do seu contido, xa que serve de ponte entre o que podemos denominar unha "visión de usuario" das linguaxes informáticas, representada pola programación estándar, e unha "visión xerativa" destas, na que o alumno constrúe e adecúa unha linguaxe de programación en atención aos seus requirimentos. Finalmente, transmítese tamén ao alumno unha visión formal dos fundamentos propios da ciencia da computación.			
Plan de continxencia	<p>1. Modificacións nos contidos: Ningunha.</p> <p>2. Metodoloxías: *Metodoloxías docentes que se manteñen: Todas.</p> <p>*Metodoloxías docentes que se modifican: En caso de necesidade, todas as metodoloxías empregadas poderían aplicarse de modo non presencial.</p> <p>3. Mecanismos de atención personalizada ao alumnado: Atención continuada en Teams, Moodle e correo electrónico.</p> <p>4. Modificacións na avaliación: Non son necesarias.</p> <p>*Observacións de avaliación: Ningunha.</p> <p>5. Modificacións da bibliografía ou webgrafía: Ningunha.</p>			

Competencias / Resultados do título	
Código	Competencias / Resultados do título

Resultados da aprendizaxe			
Resultados de aprendizaxe			Competencias / Resultados do título
Coñecer en profundidade a estrutura e función dos sistemas de descrición e recoñecemento de linguaxes formais.	A39	B6	C7
	A40		



Estudar os conceptos, modelos e técnicas relacionados con estas cuestións.	A39 A40	B6	C7
Coñecer as estruturas de datos e os algoritmos utilizados para implementar os distintos modelos de recoñecemento de linguaxes formais, así como os seus posibles dominios de aplicación práctica.	A41	B6	C6 C7
Realizar implementacións destes modelos nalgún deses dominios.	A41	B1 B2 B3	C6
Sintetizar todos os conceptos estudados en ideas concretas que permitan comprender mellor os fundamentos da computación.	A39	B6	C7
Perfeccionar as habilidades para realizar futuros traballos de análises, deseño e programación.	A40 A41	B1 B2 B3	C6
Considerar a integración das técnicas e estruturas estudadas aquí noutros dominios de aplicación.	A40 A41	B1 B2 B3 B8	C6

Contidos	
Temas	Subtemas
Preliminares sobre linguaxes formais	Alfabetos, palabras e linguaxes Linguaxes regulares e expresións regulares Autómatas finitos
Linguaxes independentes do contexto e autómatas de pila	Gramáticas regulares Gramáticas regulares e linguaxes regulares Gramáticas independentes do contexto Árbores de derivación e ambigüidade Simplificación de gramáticas independentes do contexto Propiedades das linguaxes independentes do contexto Algoritmos de análise sintáctico Autómatas de pila Forma normal de Greibach
Máquinas de Turing	Definicións básicas Máquinas de Turing como aceptadoras de linguaxes Construción de máquinas de Turing Modificacións das máquinas de Turing Máquina de Turing universal
Linguaxes recursivamente enumerables	Linguaxes aceptadas por máquinas de Turing Linguaxes regulares e independentes do contexto como linguaxes recursivas Propiedades das linguaxes recursivas e recursivamente enumerables Gramáticas non restrinxidas e linguaxes recursivamente enumerables Linguaxes sensíbeis ao contexto e a xerarquía de Chomsky
Resolubilidade	O problema da parada O problema de correspondencia de Post Problemas non decidíbeis en linguaxes independentes do contexto
Computabilidade	Fundamentos da teoría de funcións recursivas Alcance das funcións recursivas primitivas Funcións recursivas parciais O poder das linguaxes de programación

## Planificación



Metodoloxías / probas	Competencias / Resultados	Horas lectivas (presenciais e virtuais)	Horas traballo autónomo	Horas totais
Sesión maxistral	A39 A40 B8 C6 C7	18	36	54
Prácticas de laboratorio	A40 A41 B1 B2 B3 B6 B8 C6	13	26	39
Proba de resposta breve	A39 A40 B1 C6 C7	3	6	9
Solución de problemas	B1 B3 B6	4	20.5	24.5
Proba obxectiva	A39 A40 B1 C6 C7	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	A técnica que mellor se adapta á impartición dos contidos teóricos desta materia está constituída polas clases maxistrais. Nelas, faremos un uso intensivo da lousa virtual e das transparencias, de modo que o ritmo de exposición de conceptos por parte do profesor e o de asimilación dos mesmos por parte do alumno sexan o máis acordes posible.
Prácticas de laboratorio	Estas prácticas serán utilizadas para implementar nalgunha linguaxe de programación os algoritmos máis destacados, de entre todos aqueles que fosen presentados nas sesións teóricas.
Proba de resposta breve	Realizaranse controles ao final de cada bloque temático, que permitirán ao profesor coñecer o grao de asimilación da materia por parte dos alumnos, e modificar a estratexia docente se é necesario.
Solución de problemas	Poranse a disposición dos alumnos unha serie de boletíns de exercicios, correspondentes aos bloques temáticos do programa da materia. Os alumnos deberán entregar ao profesor as súas solucións persoais a estes exercicios. O profesor deberá corrixilas, avalialas e comentalas durante polo menos unha sesión.
Proba obxectiva	Implementarase baixo a forma dun exame final escrito.

Atención personalizada	
Metodoloxías	Descrición
Prácticas de laboratorio	Dado o carácter personalizado das prácticas de laboratorio e das titorías, estas actividades non deben dedicarse a estender os contidos con novos conceptos, senón a aclarar os conceptos xa expostos.  O profesor debe ademais utilizalas como unha interacción que lle permita extraer conclusións respecto ao grao de asimilación da materia por parte dos alumnos.  Desta maneira, poderá desenvolver as clases maxistrais e o resto de actividades non personalizadas atendendo ao progreso dos alumnos nas capacidades de comprensión e asimilación dos contidos impartidos, compaxinando o avance xeral da clase cunha atención específica a aqueles alumnos que presenten maiores dificultades na tarefa da aprendizaxe e cun apoio adicional a aqueles outros que presenten maior soltura e desexen ampliar coñecementos.

Avaliación			
Metodoloxías	Competencias / Resultados	Descrición	Cualificación
Proba obxectiva	A39 A40 B1 C6 C7	Exame final escrito. (***)	0
Solución de problemas	B1 B3 B6	Boletíns de exercicios e controles dos mesmos.	10
Proba de resposta breve	A39 A40 B1 C6 C7	Controles con cuestións teóricas e prácticas ao final de cada bloque temático. (**)	60



Prácticas de laboratorio	A40 A41 B1 B2 B3 B6 B8 C6	Implementación de algoritmos nalgunha linguaxe de programación e resolución de problemas. (*)	30
Outros			

### Observacións avaliación

(\*) Nas prácticas de laboratorio, requírese que o alumno obtenga unha nota mínima de 3 puntos (sobre 10).

(\*\*) A materia dividirase en tres bloques temáticos. Ao final de cada bloque temático, realizarase un control con cuestións teóricas e prácticas. Cada control poderá consolidar ata un 20% da cualificación. A porcentaxe correspondente aos controles non superados pasará a computarse na proba obxectiva (examen final). Os alumnos que superen os tres controles, non terán que realizar o exame final.

(\*\*\*) No caso de ter que realizar o exame final, requírese que o alumno obtenga unha nota mínima de 3 puntos (sobre 10).

Os alumnos a tempo parcial terán consideracións adecuadas á súa situación.

### Fontes de información

<b>Bibliografía 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>Bibliografía 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>

### Recomendacións

#### Materias que se recomenda ter cursado previamente

Programación I/614G01001  
Matemática Discreta/614G01004  
Programación II/614G01006  
Álgebra/614G01010  
Algoritmos/614G01011  
Paradigmas de Programación/614G01014

#### Materias que se recomenda cursar simultaneamente

#### Materias que continúan o temario

Representación do Coñecemento e Razoamento Automático/614G01036  
Recuperación da Información/614G01040  
Deseño das Linguaxes de Programación/614G01065  
Procesamento de Linguaxes/614G01067

### 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