



| Teaching Guide      |   |        |                                     |           |
|---------------------|---|--------|-------------------------------------|-----------|
| Identifying Data    |   |        |                                     | 2019/20   |
| Subject (*)         | Theoretical Computer Science  |        | Code                                | 614G01039 |
| Study programme     | Grao en Enxeñaría Informática   |        |                                     |           |
| Descriptors         |   |        |                                     |           |
| Cycle               | Period  | Year   | Type                                | Credits   |
| Graduate            | 2nd four-month period   | Third  | Optional                            | 6         |
| Language            | Spanish/Galician  |        |                                     |           |
| Teaching method     | Face-to-face  |        |                                     |           |
| Prerequisites       |   |        |                                     |           |
| Department          | Ciencias da Computación e Tecnoloxías da Información/Computación  |        |                                     |           |
| Coordinador         | Graña Gil, Jorge  | E-mail | jorge.grana@udc.es                  |           |
| Lecturers           | Graña Gil, Jorge<br>Novo Bujan, Jorge   | E-mail | jorge.grana@udc.es<br>j.novo@udc.es |           |
| Web                 | moodle.udc.es   |        |                                     |           |
| General description | 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 requisitos. Finalmente, transmítense tamén ao alumno unha visión formal dos fundamentos propios da ciencia da computación. |        |                                     |           |

| Study programme competences |   |
|-----------------------------|---|
| Code                        | Study programme competences   |
| A39                         | Capacidade para ter un coñecemento profundo dos principios fundamentais e modelos da computación, e saber aplicalos para interpretar, seleccionar, valorar, modelar, e crear novos conceptos, teorías, usos e desenvolvimentos tecnolóxicos relacionados coa informática. |
| A40                         | Capacidade para coñecer os fundamentos teóricos das linguaxes de programación e as técnicas de procesamento léxico, sintáctico e semántico asociadas, e saber aplicalas para a creación, o deseño e o procesamento de linguaxes.  |
| A41                         | Capacidade para avaliar a complexidade computacional dun problema, coñecer estratexias algorítmicas que poidan conducir á súa resolución e recomendar, desenvolver e implementar aquela que garanta o mellor rendemento de acordo cos requisitos establecidos.            |
| B1                          | Capacidade de resolución de problemas   |
| B2                          | Traballo en equipo  |
| B3                          | Capacidade de análise e síntese   |
| B6                          | Toma de decisións   |
| B8                          | Capacidade de traballar nun equipo interdisciplinar   |
| C6                          | Valorar críticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrentarse.   |
| C7                          | Asumir como profesional e cidadán a importancia da aprendizaxe ao longo da vida.  |

| Learning outcomes   |  |            |                             |
|---|--|------------|-----------------------------|
| Learning outcomes   |  |            | Study programme competences |
| Coñecer en profundidade a estrutura e función dos sistemas de descripción e recoñecemento de linguaxes formais.   |  | A39<br>A40 | B6<br>C7                    |
| Estudar os conceptos, modelos e técnicas relacionados con estas cuestiós.   |  | A39<br>A40 | B6<br>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<br>C6<br>C7              |
| Realizar implementacións destes modelos nalgún deses dominios.  |  | A41        | B1<br>B2<br>B3<br>C6        |



|   |            |                      |    |
|---|------------|----------------------|----|
| Sintetizar todos os conceptos estudiados 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<br>A41 | B1<br>B2<br>B3       | C6 |
| Considerar a integración das técnicas e estruturas estudiadas aquí noutros dominios de aplicación.                        | A40<br>A41 | B1<br>B2<br>B3<br>B8 | C6 |

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

| Planning                       |                              |                      |                               |             |
|--------------------------------|------------------------------|----------------------|-------------------------------|-------------|
| Methodologies / tests          | Competencies                 | Ordinary class hours | Student?s personal work hours | Total hours |
| Guest lecture / keynote speech | A39 A40 B8 C6 C7             | 18                   | 36                            | 54          |
| Laboratory practice            | A40 A41 B1 B2 B3 B6<br>B8 C6 | 13                   | 26                            | 39          |
| Short answer questions         | A39 A40 B1 C6 C7             | 3                    | 6                             | 9           |
| Problem solving                | B1 B3 B6                     | 4                    | 20.5                          | 24.5        |



|   |                  |     |    |     |
|---|------------------|-----|----|-----|
| Objective test  | A39 A40 B1 C6 C7 | 3   | 16 | 19  |
| Personalized attention  |                  | 4.5 | 0  | 4.5 |
| (*)The information in the planning table is for guidance only and does not take into account the heterogeneity of the students. |                  |     |    |     |

| Methodologies                  |  |
|--------------------------------|--|
| Methodologies                  | Description  |
| Guest lecture / keynote speech | A técnica que mellor se adapta á impartición dos contidos teóricos desta materia está constituída polas clases magistrais. Nelas, faremos un uso intensivo da lousa 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. |
| Laboratory practice            | As prácticas de laboratorio terán horas de laboratorio reservadas, con computadores a disposición dos alumnos. Estas horas serán utilizadas para implementar nalgúnha linguaxe de programación os algoritmos más destacados, de entre todos aqueles que fosen presentados nas sesións teóricas.  |
| Short answer questions         | 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.   |
| Problem solving                | 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, avaliarlas e comentalas durante polo menos unha sesión na aula.                     |
| Objective test                 | Implementarase baixo a forma dun exame final escrito.  |

| Personalized attention |  |
|------------------------|--|
| Methodologies          | Description  |
| Laboratory practice    | Dado o carácter personalizado das prácticas de laboratorio e das tutorías, estas actividades non deben dedicarse a estender os contidos con novos conceptos, senón a aclarar os conceptos xa expostos.<br><br>O profesor debe ademais utilizaras como unha interacción que lle permita extraer conclusións respecto ao grao de asimilación da materia por parte dos alumnos.<br><br>Desta maneira, poderá desenvolver as clases magistrais 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. |

| Assessment             |                              |   |               |
|------------------------|------------------------------|---|---------------|
| Methodologies          | Competencies                 | Description   | Qualification |
| Objective test         | A39 A40 B1 C6 C7             | Exame final escrito. (***)  | 0             |
| Problem solving        | B1 B3 B6                     | Boletíns de exercicios e controles dos mesmos.  | 10            |
| Short answer questions | A39 A40 B1 C6 C7             | Controles con cuestións teóricas e prácticas ao final de cada bloque temático. (**)           | 60            |
| Laboratory practice    | A40 A41 B1 B2 B3 B6<br>B8 C6 | Implementación de algoritmos nalgúnha linguaxe de programación e resolución de problemas. (*) | 30            |
| Others                 |                              |   |               |

## Assessment comments



(\*) 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.

#### Sources of information

|               |  |
|---------------|--|
| Basic         | <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>             |
| Complementary | <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> |

#### Recommendations

##### Subjects that it is recommended to have taken before

Programming I/614G01001

Discrete Mathematics/614G01004

Programming II/614G01006

Algebra/614G01010

Algorithms/614G01011

Programming Paradigms/614G01014

##### Subjects that are recommended to be taken simultaneously

##### Subjects that continue the syllabus

Knowledge Representation and Automatic Reasoning/614G01036

Information Retrieval/614G01040

Programming Language Design/614G01065

Language Processing/614G01067

#### Other comments

(\*)The teaching guide is the document in which the URV publishes the information about all its courses. It is a public document and cannot be modified. Only in exceptional cases can it be revised by the competent agent or duly revised so that it is in line with current legislation.