

## Teoría de autómatas y lenguajes formales

FICHA DE MATERIA				
<b>Denominación</b>	Computación			
Asignaturas				
Nombre	Tipo	Curso	Cuatrimestre	Nº Créditos
Informática teórica	OPT	3º	1º	6
Teoría de autómatas y lenguajes formales	OPT	3º	1º	6
Procesadores de lenguajes	OPT	3º	2º	6
Informática gráfica y visualización	OPT	3º	2º	6
<b>Competencias que se adquirirán</b>	CB-01, CB-03, CB-04, CC-01, CC-02, CC-06			
<b>Descripción de los contenidos</b>	<p>Los contenidos cubrirán la teoría y la práctica de los principales fundamentos de la computación, como son el estudio y construcción de autómatas, la construcción de procesadores de lenguajes (compiladores o intérpretes), el estudio de la complejidad de problemas complejos y la aplicación de varios de estos conceptos a problemas concretos como la informática gráfica.</p> <p>Los contenidos de las asignaturas de la materia se corresponden con algunos de los contenidos descritos por el curricula de la ACM ("<i>Programming Languages (PL)</i>"), especialmente los de la asignatura de Procesadores de lenguajes y la de Teoría de Autómatas.</p> <p>La asignatura de Teoría de Autómatas cubrirá los siguientes contenidos:</p> <ul style="list-style-type: none"> <li>– Lenguajes formales</li> <li>– Gramáticas formales</li> <li>– Autómatas finitos</li> <li>– Autómatas de pila</li> </ul> <p>Todas estas asignaturas cubren los contenidos básicos necesarios para alcanzar las competencias CC-01, CC-02 y CC-06,</p>			
<b>Resultados de aprendizaje</b>	<ul style="list-style-type: none"> <li>• Reconocer los modelos formales que sostienen la teoría del procesamiento de lenguajes, tales como las expresiones regulares y gramáticas independientes del contexto.</li> <li>• Saber construir autómatas finitos.</li> <li>• Saber construir máquinas de Turing básicas.</li> <li>• Saber discernir cuando un problema es intratable.</li> <li>• Saber discernir cuando un problema es indecidible.</li> </ul>			

- Describir las distintas fases y algoritmos utilizados en la traducción y generación de código desde el programa fuente al ejecutable, incluidas las consideraciones en la traducción de código dependiente e independiente de la máquina.
- Introducir los conceptos de optimización de código, incluyendo las distintas posibilidades en la elección de código intermedio y fases de optimización.
- Introducir los conceptos de tipos, ámbito y su comprobación (compatibilidad), en el procesamiento de lenguajes de programación.
- Ser capaz de utilizar herramientas de construcción de procesadores de lenguajes.
- Comprender y saber utilizar las técnicas matemáticas y algorítmicas necesarias para la representación óptima de objetos en un ordenador.
- Diseñar y construir modelos que representan la información para soportar la creación y visualización de imágenes.

El desarrollo del módulo se realizará de manera no presencial, utilizando el entorno virtual y la acción tutorial síncrona y asíncrona necesaria.

Como metodología general a aplicar para la adquisición de cada competencia, el estudiante deberá estudiar los materiales de cada tema apoyado por las tutorías necesarias, realizando las consultas bibliográficas que necesite. Una vez adquiridos los conceptos teóricos, el estudiante deberá realizar y comentar los ejercicios propuestos, para pasar a realizar prácticas individuales o grupales de mayor envergadura debidamente documentadas. La evaluación de cada competencia tendrá carácter formativo, pudiéndose evaluar más de una competencia a la vez.

**Actividades formativas**

Estudio personal	216
Lecturas complementarias dirigidas	36
Clases, conferencias o técnicas expositivas	36
Elaboración de trabajos individuales	108
Elaboración de trabajos en grupo	36
Tutoría individual	36
Participación en foros/medios colaborativos	36
Test de autoevaluación	36
Prácticas de laboratorio	180

El desarrollo del módulo se realizará de manera no presencial, utilizando el entorno virtual y la acción tutorial síncrona y asíncrona necesaria (TUT/FO). Por ello, no se necesita un laboratorio instalado desde la UNIR, sino de simuladores instalados por cada alumno en su equipo personal Informática gráfica y

	<p>visualización</p> <p>Se dispondrá un laboratorio que permita el diseño y despliegue de prácticas enfocadas en recuperación y visualización de información, así como en su representación gráfico y posterior análisis, con especial énfasis en datasets de gran volumen y técnicas de Visual LearningAnalytics. Se seguirán las pautas proporcionadas por la asociación LAK ( <a href="http://lak12.sites.olt.ubc.ca/">http://lak12.sites.olt.ubc.ca/</a>) y entidades relacionadas</p>
<p><b>Sistema de evaluación de la adquisición de competencias</b></p>	<p><u>Evaluación continua (progresiva):</u>  10%: Resolución de trabajos, proyectos y casos.  20%: Prácticas de laboratorio  5%: Participación en foros y otros medios colaborativos  5%: Lecturas complementarias  <b>Total de evaluación continua: 40 %</b></p> <p><u>Evaluación final (presencial):</u>  Prueba de evaluación final. Hay que aprobar el examen final, que será físicamente presencial para asegurar la identificación del estudiante, para que se tome en consideración la calificación de la evaluación continua.  <b>Total de evaluación final: 60%</b></p>