

Guía docente de la asignatura

## Modelos de Computación (4961131)



Fecha de aprobación: 27/06/2024

<b>Grado</b>	Grado en Ingeniería Informática (Ceuta)	<b>Rama</b>	Ingeniería y Arquitectura
--------------	---	-------------	---------------------------

<b>Módulo</b>	Formación Específica de Rama	<b>Materia</b>	Programación e Ingeniería del Software
---------------	------------------------------	----------------	--

<b>Curso</b>	3º	<b>Semestre</b>	1º	<b>Créditos</b>	6	<b>Tipo</b>	Obligatoria
--------------	----	-----------------	----	-----------------	---	-------------	-------------

### PRERREQUISITOS Y/O RECOMENDACIONES

Los estudiantes no tendrán que tener asignaturas, materias o módulos aprobados como requisito indispensable para cursar el módulo. No obstante, se recomienda la superación de los contenidos y adquisición de competencias de las materias de formación básica.

### BREVE DESCRIPCIÓN DE CONTENIDOS (Según memoria de verificación del Grado)

- Introducción a la Computación.
- Autómatas Finitos y Expresiones Regulares.
- Gramáticas Libres del Contexto.
- Autómatas con PILA.
- Lenguajes Libres del Contexto Determinísticos.
- Lenguajes Dependientes del Contexto.

### COMPETENCIAS ASOCIADAS A MATERIA/ASIGNATURA

#### COMPETENCIAS GENERALES

- CG09 - Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

#### COMPETENCIAS ESPECÍFICAS

- CE12 - Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.



## RESULTADOS DE APRENDIZAJE (Objetivos)

- Usar con soltura el lenguaje matemático, comprender y generar demostraciones relacionadas con los contenidos.
- Clasificar los lenguajes según el tipo de gramática o máquina requerido.
- Conocer las relaciones de jerarquía entre clases de lenguajes.
- Analizar cuál es el lenguaje generado por una gramática, descrito por una expresión regular o reconocido por una máquina teórica.
- Diseñar autómatas finitos, con pila o máquinas de Turing como modelos para resolver problemas relacionados con el reconocimiento de lenguajes.
- Conocer la relación entre lenguajes y entre máquinas, así como la equivalencia entre distintos tipos de máquinas teóricas y la equivalencia entre máquinas y gramáticas.
- Aplicar algoritmos para realizar conversiones entre especificaciones igual de potentes para un lenguaje.
- Evaluar cuál es la máquina más adecuada para reconocer un lenguaje, atendiendo a la dificultad de tratamiento computacional.
- Conocer los límites de los procesos computacionales y la implicación práctica de la irresolubilidad o intratabilidad de un problema abstracto.
- Conocer la relación entre problemas, funciones y algoritmos, así como la equivalencia entre distintos modelos de computación.
- Aplicar diversos modelos de computación para el cálculo de funciones numéricas o con cadenas.

## PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

### TEÓRICO

#### Tema 1: Introducción a la computación.

- Conceptos Elementales
- Modelos de Cálculo
- La noción de Gramática Generativa
- Operaciones con Lenguajes

#### Tema 2: Autómatas Finitos y Expresiones Regulares

- Autómatas Finitos Deterministas
- Autómatas No-Deterministas
- Expresiones Regulares
- Gramáticas Regulares

#### Tema 3: Propiedades de los Conjuntos Regulares

- Lema de Bombeo y Aplicaciones
- Minimización de Autómatas

#### Tema 4: Gramáticas Independientes del Contexto

- Introducción
- Árboles de Derivación. Ambigüedad
- Simplificación de Gramáticas
- Formas Normales

#### Tema 5: Autómatas con Pila

- Definiciones
- Autómatas con Pila y Lenguajes Libres del Contexto
- Autómatas con Pila Deterministas

#### Tema 6. Propiedades de los Lenguajes Independientes del Contexto.

- Lema de Bombeo.
- Propiedades de Clausura.



- Algoritmos.
- Tema 7. Máquinas de Turing
- Máquinas de Turing
  - Lenguajes recursivos y recursivamente enumerables
  - El problema de la parada para máquinas de Turing

## PRÁCTICO

### Prácticas

- Práctica 1: Resolución de problemas relacionados con Autómatas Finitos y Expresiones Regulares.
- Práctica 2: Resolución de problemas relacionados con Gramáticas Independientes del Contexto y Autómatas con Pila.
- Práctica 3: Resolución de Problemas relacionados con Máquinas de Turing.

### Seminarios

- Seminario 1: LEX
- Seminario 2: Simulación de autómatas y gramáticas

## BIBLIOGRAFÍA

### BIBLIOGRAFÍA FUNDAMENTAL

- Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2010). Introducción a la teoría de autómatas, lenguajes y computación / John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (3a ed.). Pearson.
- Gopalakrishnan, G. (2020). Automata and Computability: A Programmer's Perspective. CRC Press.
- Weber, R. (2012). Computability Theory. American Mathematical Soc.
- Carroll, J., & Long, D. (1989). Theory of finite automata with an introduction to formal languages. Prentice-Hall International.

### BIBLIOGRAFÍA COMPLEMENTARIA

- Aho, A. V., & Ullman, J. D. (1992). Foundations of computer science. Computer Science Press.
- Alfonseca Cubero, E., Alfonseca, M., & Moriyón Salomón, R. (2007). Teoría de autómatas y lenguajes formales / Enrique Alfonseca Cubero, Manuel Alfonseca Moreno, Roberto Moriyón Salomón. McGraw-Hill.
- Beckmann, A., Mitrana, V., & Soskova, M. (2015). Evolving Computability [electronic resource] : 11th Conference on Computability in Europe, CiE 2015, Bucharest, Romania, June 29-July 3, 2015. Proceedings / edited by Arnold Beckmann, Victor Mitrana, Mariya Soskova. (A. Beckmann, V. Mitrana, & M. Soskova, Eds.; 1st ed. 2015.). Springer International Publishing. <https://doi.org/10.1007/978-3-319-20028-6>.
- Book, R.V., Otto, F. (1993). String rewriting systems. Springer-Verlag, Nueva York.
- Brookshear, J. G., & Morales Peake, E. (1993). Teoría de la computación : lenguajes formales, autómatas y complejidad. Addison Wesley Iberoamericana.
- Cohen, D. I. A. (1991). Introduction to computer theory / Daniel I.A. Cohen (rev. ed). John Wiley.
- Cutland, N. (1994). Computability : an introduction to recursive function theory. Cambridge University Press.



- Davis, M. D., Sigal, R., & Weyuker, E. J. (1994). Computability, complexity, and languages : fundamentals of theoretical computer science / Martin D. Davis, Ron Sigal, Elaine J. Weyuker (2nd. ed). Academic Press.
- Grune, D., & Jacobs, C. J. H. (2008). Parsing Techniques [electronic resource] : A Practical Guide / by Dick Grune, Ceriel J.H. Jacobs. (2nd ed. 2008.). Springer New York. <https://doi.org/10.1007/978-0-387-68954-8>
- Harrison, M. A. (1978). Introduction to formal language theory. Addison Wesley.
- Howie, J. M. (1991). Automata and languages / John M. Howie. Clarendon Press.
- Kelley, D., & Díez Platas, M. L. (2001). Teoría de autómatas y lenguajes formales / Dean Kelly ; traducción M<sup>a</sup> Luísa Díez Platas. Prentice-Hall.
- Lewis, H. R., & Papadimitriou, C. H. (1981). Elements of the theory of computation. Prentice-Hall.
- Plotkin, B. I., Gvaramija, A. A., & Greenglaz, L. J. (1992). Algebraic structures in automata and databases theory. World Scientific.
- Revesz, G. E. (1991). Introduction to formal languages / György E. Révész. Dover.
- Sommaruga, G., & Strahm, T. (2015). Turing's Revolution [electronic resource] : The Impact of His Ideas about Computability / edited by Giovanni Sommaruga, Thomas Strahm. (G. Sommaruga & T. Strahm, Eds.; 1st ed. 2015.). Springer International Publishing. <https://doi.org/10.1007/978-3-319-22156-4>.
- Sudkamp, T. A. (2007). Languages and machines : an introduction to the theory of computer science / Thomas A. Sudkamp (3rd. ed.). Pearson Education.

## ENLACES RECOMENDADOS

- [Herramientas para la enseñanza de autómatas y gramáticas en Java](#) (por Susan H. Rodger, Duke University)
- [Página del libro de Hopcroft, Motwani, Ullman con material adicional y soluciones de ejercicios](#)
- [Aplicaciones de las expresiones regulares](#)
- [Cuatro aplicaciones básicas de expresiones regulares](#)
- [Programa para trabajar con expresiones regulares](#)

## METODOLOGÍA DOCENTE

- MD01 - Lección Magistral (Clases Teóricas-Expositivas)
- MD02 - Actividades Prácticas (Resolución de Problemas, Resolución de Casos Prácticos, Desarrollo de Proyectos, Prácticas en Laboratorio, Taller de Programación, Aula de Informática, Prácticas de Campo).
- MD03 - Seminarios (Debates, Demos, Exposición de Trabajos Tutelados, Conferencias, Visitas Guiadas, Monografías).
- MD04 - Actividades no presenciales Individuales.
- MD05 - Actividades no presenciales Grupales.
- MD06 - Tutorías Académicas.

## EVALUACIÓN (instrumentos de evaluación, criterios de evaluación y porcentaje sobre la calificación final)

### EVALUACIÓN ORDINARIA



Para la parte teórica se realizará un examen. La ponderación de este bloque es del 50%. Para la parte práctica se realizará resolución de problemas o el desarrollo de proyectos (individuales o en grupo), entregas de informes/memorias solicitados a los estudiantes, entrevistas personales con los alumnos, sesiones de evaluación y participación. La ponderación de este bloque es del 50%, compuesto por:

- Notas de problemas: 30%.
- Trabajos Personales, participación en clase y/o Exposición: 20%

La calificación global corresponderá por tanto a la puntuación ponderada de los diferentes aspectos y actividades que integran el sistema de evaluación. Por tanto, el resultado de la evaluación será una calificación numérica obtenida mediante la suma ponderada de las calificaciones correspondientes a una parte teórica, una parte práctica y, en su caso, una parte relacionada con el trabajo autónomo de los alumnos, los seminarios impartidos. Se exigirá una nota mínima de 3,5 sobre 10 puntos tanto en la parte práctica como en la teórica para realizar la suma ponderada. Si un estudiante no supera ese mínimo en una de sus partes, la nota máxima final no podrá superar el valor de 4,5.

Todo lo relativo a la evaluación se registrará por la normativa sobre planificación docente y organización de exámenes vigente en la Universidad de Granada.

### EVALUACIÓN EXTRAORDINARIA

En la convocatoria extraordinaria se realizará un examen único que se valorará entre 0 y 10 y tal que incluirá pruebas relativas a la parte teórica y práctica (en un porcentaje del 50% para teoría y 50% práctica)

### EVALUACIÓN ÚNICA FINAL

De acuerdo a lo establecido en la Normativa de evaluación y de calificación de los estudiantes de la Universidad de Granada vigente, la evaluación será preferentemente continua. No obstante, el estudiante que no pueda acogerse a dicho sistema por motivos laborales, estado de salud, discapacidad, programas de movilidad o cualquier otra causa debidamente justificada podrá acogerse a la evaluación única final. Para ello deberá solicitarlo al Director del Departamento o al Coordinador del Máster en las dos primeras semanas de impartición de la asignatura o, excepcionalmente, en las dos primeras semanas tras la matriculación en la asignatura.

Esta modalidad de evaluación se realizará en un único acto académico en la fecha establecida por el Centro y consistirá en un examen escrito (evaluado de 0 a 10) que incluirá preguntas tanto de tipo teórico como práctico que garanticen que el alumno ha adquirido la totalidad de las competencias descritas en esta misma guía docente.

### INFORMACIÓN ADICIONAL

Información de interés para estudiantado con discapacidad y/o Necesidades Específicas de Apoyo Educativo (NEAE): [Gestión de servicios y apoyos \(https://ve.ugr.es/servicios/atencion-social/estudiantes-con-discapacidad\)](https://ve.ugr.es/servicios/atencion-social/estudiantes-con-discapacidad).

