

Curso Académico: ( 2016 / 2017 )

Fecha de revisión: 29-04-2016

Departamento asignado a la asignatura: Departamento de Informática

Coordinador/a: SANCHIS DE MIGUEL, MARIA ARACELI

Tipo: Obligatoria Créditos ECTS : 6.0

Curso : 3 Cuatrimestre : 1

**REQUISITOS (ASIGNATURAS O MATERIAS CUYO CONOCIMIENTO SE PRESUPONE)**

Programación

Estructuras de Datos y Algoritmos

**OBJETIVOS**

El objetivo de este curso es que el estudiante adquiera las siguientes competencias:

**COMPETENCIAS EURO-ACE****1.- Competencias Básicas:**

CB5: Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

**2.- Competencias generales y transversales.**

CGB3. Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería

**3.- Competencias comunes a la rama de Informática.**

CECRI6. 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.

CECRI15. Conocimiento y aplicación de los principios fundamentales y técnicas básicas de los sistemas inteligentes y su aplicación práctica.

**COMPETENCIAS ABET****1.-Competencias Transversales/Genéricas**

- Capacidad de análisis y síntesis. PO: a,c,e,g
- Resolución de problemas. PO: a,c,e
- Razonamiento crítico. PO: a,c,e,g,h,k
- Trabajo en equipo. PO: g
- Comunicación escrita. PO: g
- Automatizar procesos. PO: a,c,e,h,k

**2.Competencias Específicas del Aprendizaje****a.Cognitivas (Saber).**

PO: a

- Conocer las teorías formales para la descripción de lenguajes.
- Conocer el concepto de gramática formal y sus tipos, así como los tipos de lenguajes.
- Conocer el concepto de autómata finito como reconocedor de lenguajes regulares.
- Conocer el concepto de expresión regular como descripción de un lenguaje regular.
- Conocer el concepto de autómata a pila para el reconocimiento de lenguajes independientes del contexto
- Comprender la correspondencia entre gramáticas, lenguajes y reconocedores.
- Conocer los fundamentos y el funcionamiento de la máquina de Turing y los distintos tipos de máquina de Turing.
- Conocer el concepto de complejidad computacional. Conocer los métodos usados para calcular la complejidad computacional de un algoritmo
- Conocer el concepto de clases de problemas P y NP.
- Conocer cuáles son las capacidades y límites de la computación.

**b.Procedimentales**

- Capacitar al alumno para evaluar cómo abordar un problema de reconocimiento de palabras para una gramática dada.

PO: c,e,g

- Plantear correctamente las distintas fases para la construcción de un reconocedor, desde la descripción de la gramática hasta el diseño del autómata

PO: a

- Combinar y extrapolar los conocimientos adquiridos para la construcción de un reconocedor léxico ó sintáctico de una gramática, a partir de los conocimientos sobre reconocedores.

PO: a,c,e,g,h,k

- Capacidad de valorar la eficiencia de un autómata determinado para el reconocimiento de un lenguaje concreto (valorar si el autómata es mínimo).

PO: a,c,e,g

- Aplicación práctica de los fundamentos teóricos de los modelos de dispositivos de computación/cálculo expuestos (Gramáticas, Autómatas Finitos, Autómatas a Pila y Máquinas de Turing) para la resolución de problemas de cómputo y cálculo.

PO: a,e,g

- Capacidad para determinar el orden de complejidad de un algoritmo, un autómata y una máquina de Turing

PO: a,e,g

- Capacidad para transformar enunciados informales a enunciados formales.

PO: a,e,g

### 3.-Actitudinales (Ser)

- Capacidad para analizar los problemas y sus soluciones.

- Preocupación por la calidad.

## DESCRIPCIÓN DE CONTENIDOS: PROGRAMA

### 1.Introducción a la teoría de Autómatas y Lenguajes Formales.

1.1. Por qué de la Teoría de Autómatas. Historia y Origen

1.2. Relación con otras Áreas de Conocimiento.

1.2. Máquinas, Lenguajes y Algoritmos.

### 2.- Teoría de Autómatas

2.1.Introducción y Definiciones.

2.2 Modelo Matemático de un Autómata

2.3 Autómatas y Algoritmos

2.4 Autómatas discretos, continuos e híbridos.

2.5 Clases de Autómata

### 3.Autómatas Finitos

3.1. Definición y representación de Autómatas Finitos Deterministas (AFD)

3.2. AFD como reconocedores de lenguajes

3.3. Equivalencia y minimización de AFD

3.4. Teoremas sobre AFD

3.5. Definición y representación de Autómatas Finitos No Deterministas (AFND)

3.6. Lenguaje aceptado por un AFND

3.7. Equivalencia entre AFD y AFND

### 4.Lenguajes y Gramáticas formales.

4.1. Operaciones con Palabras. Operaciones con Lenguajes. Reglas de Derivación

4.2. Concepto de Gramática. Definición de Gramática Formal

4.3. Jerarquía de Chomsky y Gramáticas Equivalentes

4.4. Gramáticas Independientes del Contexto (Tipo 2)

4.5. Lenguaje Generado por una Gramática Tipo 2. Árboles de Derivación

4.6. Gramáticas Bien Formadas

4.7. Forma Normal de Chomsky. Forma Normal de Greibach

### 5.Lenguajes regulares.

5.1. Definición de Lenguajes regular

5.2. AFD asociado a una Gramática de Tipo 3

5.3. Expresiones Regulares. Equivalencias

5.4. Teoremas de Kleene

5.5. Ecuaciones características

5.6. Algoritmo recursivo de síntesis

5.7. Derivada de una expresión regular

## 6. Autómatas a pila.

- 6.1. Definición de Autómata a Pila (AP)
- 6.2. Movimientos y Descripciones Instantáneas en AP
- 6.3. AP por vaciado (APV) y AP por estados finales (APF)
- 6.4. Lenguaje aceptado por un AP: equivalencia APV y APF
- 6.5. Construcción de APV a partir de una Gramática Tipo 2
- 6.6. Construcción de una Gramática Tipo 2 a partir de AP

## 7. Máquina de Turing

- 7.1. Definición de la Máquina de Turing
- 7.2. Variaciones de la Máquina de Turing
- 7.3. Máquina de Turing Universal

## 8. Complejidad Computacional

- 8.1. Teoría de la Complejidad
- 8.2. Complejidad de Algoritmos
- 8.3. Problemas P versus NP
- 8.4. Clases de Complejidad
- 8.5. Complejidad temporal
- 8.6. Teoremas de jerarquía
- 8.7. Problemas no computacionales
- 8.8. Límites de la Computación

## ACTIVIDADES FORMATIVAS, METODOLOGÍA A UTILIZAR Y RÉGIMEN DE TUTORÍAS

Clases Magistrales (contenido teórico): 1.5 ECTS.

PO: a,c,e,g

Suponen una guía para que el alumno pueda alcanzar las competencias cognitivas, así como los elementos básicos para desarrollar competencias procedimentales. Una parte de estos ECTS se corresponde con la carga de trabajo personal del alumno.

Clases Prácticas (Ejercicios, Problemas y Prácticas): 2 ECTS.

PO: a,c,e,g,h,k

Permiten desarrollar las competencias genéricas y aplicar las actitudinales. Consisten en desarrollar y resolver casos prácticos (ejercicios, problemas y prácticas) con los que además se permite alcanzar las competencias procedimentales. Una parte importante de estos ECTS se corresponde con la carga de trabajo personal del alumno.

Tutorías Colectivas:

A lo largo del curso se llevara a cabo dos tutorias colectivas.

### Realización de otras Actividades Académicas

- En presencia del profesor: 0.5 ECTS.

Resolución de pequeñas cuestiones, ejercicios, y prácticas que tendrán peso en la nota final de la asignatura. Parte de los ECTS se corresponde con el repaso de los contenidos de la materia por parte del alumno.

PO: a,c,e,g,h,k

- En ausencia del profesor: 1.5 ECTS.

Lecturas relativas al contenido de la materia, así como la realización de ejercicios, problemas y prácticas relacionadas con las clases magistrales y las clases prácticas.

PO: a,c,e,g,h,k

Examen: 0.5 ECTS.

Preparación y realización del examen, en el se evalúan el nivel alcanzado por el alumno en relación a las competencias específicas del aprendizaje.

## SISTEMA DE EVALUACIÓN

La evaluación consistirá en evaluación continua y en una prueba final.

El objetivo de la evaluación continua es ayudar a los estudiantes a monitorizar su progreso en el proceso de aprendizaje, recibiendo una retroalimentación continua del grado de asimilación de las competencias durante el curso. Así pues, cada prueba

parcial y el trabajo práctico supondrán tanto una actividad de aprendizaje como de evaluación.

La prueba final tiene como objetivo establecer el grado de asimilación el grado de adquisición de competencias cognitivas y procedimentales.

La evaluación continua supondrá el 50% de la nota final de la asignatura.

La evaluación continua consistirá en:

- Tres pruebas escritas (preguntas test, preguntas cortas y resolución de problemas),
- Cuatro trabajos prácticos en los que se empleará la herramienta JFLAP (<http://www.cs.duke.edu/csed/jflap/>).

Cada una de las tres pruebas escritas supondrá el 12% de la nota de la asignatura y los trabajos prácticos supondrán el 14% de la nota de la asignatura.

PO: a,c,e,g,h,k

El examen final (50% de la nota final de la asignatura) constará de preguntas teóricas, ejercicios y problemas prácticos.

PO: a,c,e,g

Solo se sumará la nota obtenida mediante evaluación continua si se obtiene una nota de 4 o más en el examen final.

**Peso porcentual del Examen Final:** 50

**Peso porcentual del resto de la evaluación:** 50

#### BIBLIOGRAFÍA BÁSICA

- Enrique Alfonseca Cubero, Manuel Alfonseca Cubero, Roberto Moriyón Salomón. Teoría de autómatas y lenguajes formales., McGraw-Hill (2007)..
- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. Introduction to Automata Theory, Languages, and Computation (Third Edition), Pearson Education, Pearson Addison Wesley.
- Manuel Alfonseca, Justo Sancho, Miguel Martínez Orga. Teoría de lenguajes, gramáticas y autómatas., Publicaciones R.A.E.C. ISBN: 8460560929. 1997..
- Pedro Isasi, Paloma Martínez y Daniel Borrajo. Lenguajes, Gramáticas y Autómatas. Un enfoque práctico., Addison-Wesley, (1997).
- Susan H. Rodger and Thomas W. Finley. JFLAP: An Interactive Formal Languages and Automata Package. 2006, Jones & Bartlett Publishers, Sudbury, MA. ISBN 0763738344.

#### BIBLIOGRAFÍA COMPLEMENTARIA

- Brookshear, J. Glenn. Teoría de la computación : lenguajes formales, autómatas y complejidad., Addison Wesley Iberoamericana. 1993. ISBN: 9684443846.
- Jeffrey Shallit. A Second Course in Formal Languages and Automata Theory., Cambridge University Press, September 30 2008..
- Michael Sipser. Introduction to the Theory of Computation (2nd Edition) 2006, Thomson Course Technology..
- Peter Linz An Introduction to Formal Languages and Automata. Third Edition, Jones and Bartlett Publishers. ISBN: 0763714224..
- R. Penrose La Nueva Mente del Emperador, Mondadori, 1991..