

Curso Académico: (2019 / 2020)

Fecha de revisión: 25-05-2020

Departamento asignado a la asignatura: Departamento de Informática

Coordinador/a: ALONSO WEBER, JUAN MANUEL

Tipo: Obligatoria Créditos ECTS : 6.0

Curso : 4 Cuatrimestre : 2

REQUISITOS (ASIGNATURAS O MATERIAS CUYO CONOCIMIENTO SE PRESUPONE)

Teoría de Autómatas y Lenguajes Formales

Matemática Discreta

Heurística y Optimización

OBJETIVOS

Competencias Transversales/Genericas

- Capacidad de análisis y síntesis. PO: a,e,g
 - Resolución de problemas. PO: a,e
 - Razonamiento crítico. PO: b,g
 - Trabajo en equipo. PO: b,e,g
 - Comunicación escrita. PO: g
-
- CB4 Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

Competencias Especificas del aprendizaje

a.- Cognitivas (Saber)

PO: a

- Conocer los conceptos que determinan la informática desde un punto de vista formal.
- Conocer cuales son las capacidades y límites de la Informática
- Conocer los métodos usados para calcular la complejidad computacional de un algoritmo.
- Conocer qué se puede computar de forma eficiente y las alternativas existentes.
- Conocer el concepto de Reducción de un problema a otro.

b.- Procedimentales (Saber hacer)

PO: a,b,e,g,h,i

- Determinar qué se puede computar y que no.
- Determinar si se puede computar de forma eficiente y las alternativas existentes.
- Determinar que algoritmo es más eficiente.
- Seleccionar las estructuras de datos y técnicas de programación apropiadas para diseñar un algoritmo eficiente.
- Encontrar el modelo de computación más simple para cada problema.
- Capacidad para transformar enunciados informales a enunciados formales, y al revés.

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

- 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.

- CECC1

Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.

- CECC3 Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

c.- Actitudinales (Ser)

PO: a,b,e,g,h,i

- Capacidad para analizar los problemas y sus soluciones.
- Preocupación por la calidad.
- Interés por investigar nuevas alternativas

DESCRIPCIÓN DE CONTENIDOS: PROGRAMA

1. Coste de los Procesos Computacionales.
 - 1.1 Complejidad Computacional.
 - 1.2 Coste Computacional: Programacion Estructurada
 - 1.3 Complejidad Algoritmos Recursivos.
 - 1.4 Análisis Probabilístico de la Complejidad Computacional
2. Introducción a la Teoría de la Computabilidad
 - 2.1 Definición de Problema
 - 2.2 Máquinas de Turing. Decidibilidad
 - 2.3 Clase de Problemas (Computabilidad)
3. Introducción a la Teoría de la Complejidad Computacional
 - 3.1 Relación entre problemas. Clases de Problemas
 - 3.2 Clases P, NP y NP-Completo.
 - 3.3 Clases PSpace, NPSpace.
 - 3.3 Clases NP-Hard, Exp, CoP, CoNP.
4. Otros Modelos de Computación
 - 4.1 Lambda-Calculus
 - 4.2 Modelo RAM
 - 4.3 Automatas Celulares
 - 4.4 Lindenmayer Systems
 - 4.5 Algoritmos Probabilísticos

ACTIVIDADES FORMATIVAS, METODOLOGÍA A UTILIZAR Y RÉGIMEN DE TUTORÍAS

Clases Magistrales: 1.5 ECTS.

Suponen una guía para que el alumno pueda alcanzar las competencias de cognitivas, así como los elementos básicos para las competencias procedimentales.

PO: a,b,e,g,h,i

Clases Prácticas: 1.5 ECTS.

Permiten desarrollar las competencias genéricas y aplicar las actitudinales.

Consisten en desarrollar y resolver casos prácticos en los que además permiten alcanzar las competencias procedimentales. Una parte importante de estos ECTS se corresponde con la carga de trabajo personal del alumno.

PO: a,b,e,g,h,i

Realización de Actividades Académicas

En presencia del profesor: 1.0 ECTS.

Resolución de pequeñas cuestiones y ejercicios.

Trabajos Prácticos
PO: a,e,g,

En ausencia del profesor: 1.5 ECTS.

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

PO: a,b,e,g,h,i

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

PO: a,b,e,g

SISTEMA DE EVALUACIÓN

Se realizará un examen final con el que se evalúan las competencias relacionadas con las clases magistrales y seminarios: 35% de la nota final.

Se deberá alcanzar un nota mínima (4 de 10) en el examen para poder aprobar la asignatura.

PO: a,b,e,g

Actividades académicas en presencia del profesor: 30% de la nota final.

PO: a,e,g

Evaluación de los trabajos prácticos: 35% de la nota final.

PO: a,b,e,g,h,i

Peso porcentual del Examen Final:	35
--	-----------

Peso porcentual del resto de la evaluación:	65
--	-----------

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. Introducción a la teoría de autómatas, lenguajes y computación, Addison-Wesley 2007.
- Michael Sipser. Introduction to the Theory of Computation. 2nd ed., Boston, MA: Course Technology, 2005. ISBN: 0534950973..
- S. Wolfram. Cellular Automata and Complexity., Addison-Wesley, (1996).

BIBLIOGRAFÍA COMPLEMENTARIA

- C. Papadimitriou Computational Complexity., Addison-Wesley, 1995.
- H. S. Wilf Algorithms and Complexity., Prentice-Hall, 1986.
- Jeffrey Shallit. A Second Course in Formal Languages and Automata Theory., Cambridge University Press..