

Curso Académico: (2023 / 2024)

Fecha de revisión: 09/02/2024 11:37:00

Departamento asignado a la asignatura: Departamento de Informática

Coordinador/a: SANCHIS DE MIGUEL, MARIA ARACELI

Tipo: Obligatoria Créditos ECTS : 6.0

Curso : 2 Cuatrimestre : 1

REQUISITOS (ASIGNATURAS O MATERIAS CUYO CONOCIMIENTO SE PRESUPONE)

Programación (Curso1º 1 Cuatrimestre)

Técnicas de Programación (Curso1º 2 Cuatrimestre)

RESULTADOS DEL PROCESO DE FORMACIÓN Y APRENDIZAJE

CB1. Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.

CB2. Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

CB3. Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

CB4. Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

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

CG1. Que los estudiantes sean capaces de demostrar conocimiento y comprensión de conceptos de matemáticas, estadística y computación y aplicarlos a la resolución de problemas en ciencia e ingeniería con capacidad de análisis y síntesis.

CG3. Que los estudiantes puedan resolver computacionalmente con ayuda de las herramientas informáticas más avanzadas los modelos matemáticos que surjan de aplicaciones en la ciencia, la ingeniería, la economía y otras ciencias sociales.

CG4. Que los estudiantes demuestren que pueden analizar e interpretar las soluciones obtenidas con ayuda de la informática de los problemas asociados a modelos matemáticos del mundo real, discriminando los comportamientos más relevantes para cada aplicación.

CG6. Que los estudiantes sepan buscar y utilizar los recursos bibliográficos, en soporte físico o digital, necesarios para plantear y resolver matemática y computacionalmente problemas aplicados que surjan en entornos nuevos, poco conocidos o con información insuficiente.

CE14. Que los estudiantes hayan demostrado que conocen la teoría de gramáticas, lenguajes y autómatas y que pueden aplicarla al diseño de analizadores de lenguajes de programación y de lenguajes específicos de dominio, así como que comprender el proceso de traducción de lenguajes de alto nivel y las optimizaciones más comunes que tienen lugar.

CE21. Que los estudiantes hayan demostrado que comprenden la influencia y utilidad de los fundamentos matemáticos utilizados en los lenguajes de programación funcional y el impacto de la aplicación práctica de dichos lenguajes.

RA3. Tener la capacidad de recopilar e interpretar datos e informaciones sobre las que fundamentar sus conclusiones incluyendo, cuando sea preciso y pertinente, la reflexión sobre asuntos de índole social, científica o ética en el ámbito de su campo de estudio.

RA5. Saber comunicar a todo tipo de audiencias (especializadas o no) de manera clara y precisa, conocimientos, metodologías, ideas, problemas y soluciones en el ámbito de su campo de estudio.

RA6. Ser capaces de identificar sus propias necesidades formativas en su campo de estudio y entorno laboral o profesional y de organizar su propio aprendizaje con un alto grado de autonomía en todo tipo de contextos (estructurados o no).

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 Autómatas

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 TEÓRICO-PRÁCTICAS [44 horas con un 100% de presencialidad, 1.67 ECTS]

Conocimientos que deben adquirir los alumnos. Estos recibirán las notas de clase y tendrán textos básicos de referencia para facilitar el seguimiento de las clases y el desarrollo del trabajo posterior. Se resolverán ejercicios, prácticas problemas por parte del alumno y se realizarán talleres y prueba de evaluación para adquirir las capacidades necesarias.

TUTORÍAS [4 horas con un 100% de presencialidad, 0.15 ECTS]

Asistencia individualizada (tutorías individuales) o en grupo (tutorías colectivas) a los estudiantes por parte del profesor.

TRABAJO INDIVIDUAL O EN GRUPO DEL ESTUDIANTE. [98 horas con 0% de presencialidad, 3.72 ECTS]

TALLERES Y LABORATORIOS. [8 horas con 100% de presencialidad, 0.3 ECTS]

EXAMEN FINAL. [4 horas con 100% de presencialidad, 0.15 ECTS]

Se valorarán de forma global los conocimientos, destrezas y capacidades adquiridas a lo largo del curso.

METODOLOGÍAS DOCENTES

CLASE TEORÍA. Exposiciones en clase del profesor con soporte de medios informáticos y audiovisuales, en las que se desarrollan los conceptos principales de la materia y se proporcionan los materiales y la bibliografía para complementar el aprendizaje de los alumnos.

PRÁCTICAS. Resolución de casos prácticos, problemas, etc. planteados por el profesor de manera individual o en grupo.

TUTORÍAS. Asistencia individualizada (tutorías individuales) o en grupo (tutorías colectivas) a los estudiantes por parte del profesor.

PRÁCTICAS DE LABORATORIO. Docencia aplicada/experimental a talleres y laboratorios bajo la supervisión de un tutor.

SISTEMA DE EVALUACIÓN

| | |
|--|----|
| Peso porcentual del Examen/Prueba Final: | 50 |
| Peso porcentual del resto de la evaluación: | 50 |

La evaluación consistirá en varias actividades de evaluación continua SE2 y en una prueba final SE1.

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 SE2 supondrá el 50% de la nota final de la asignatura.

La evaluación continua consistirá en:

- Dos-tres pruebas escritas (preguntas test, preguntas cortas y resolución de problemas),

Peso porcentual del Examen/Prueba Final: 50

Peso porcentual del resto de la evaluación: 50

- 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 (o 18% si son 2) y los trabajos prácticos supondrán el 14% de la nota de la asignatura.

El examen final SE1 (50% de la nota final de la asignatura) constará de preguntas teóricas, ejercicios y problemas prácticos.

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

SE1 - EXAMEN FINAL. [50 %]

En el que se valorarán de forma global los conocimientos, destrezas y capacidades adquiridas a lo largo del curso.

SE2 - EVALUACIÓN CONTINUA. [50 %]

En ella se valorarán los trabajos, presentaciones, actuación en debates, exposiciones en clase, ejercicios, prácticas y trabajo en los talleres a lo largo del curso.

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..

