

Curso Académico: (2023 / 2024)

Fecha de revisión: 09/02/2024 12:40:50

Departamento asignado a la asignatura: Departamento de Informática

Coordinador/a: ABDULLA JASSIM, HARITH AL JUMAILY

Tipo: Formación Básica Créditos ECTS : 6.0

Curso : 2 Cuatrimestre : 2

Rama de Conocimiento: Ingeniería y Arquitectura

REQUISITOS (ASIGNATURAS O MATERIAS CUYO CONOCIMIENTO SE PRESUPONE)

- Programación (Curso: 1 / Cuatrimestre: 1)
- Cálculo (Curso: 1 / Cuatrimestre: 1)

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. Tipo Abstracto de Datos
2. Tipos Abstractos de Datos lineales: pilas, colas, lista enlazadas.
3. Complejidad de Algoritmos.
4. Algoritmos recursivos.
5. Árboles
6. Grafos.
7. Divide y Vencerás.

ACTIVIDADES FORMATIVAS, METODOLOGÍA A UTILIZAR Y RÉGIMEN DE TUTORÍAS

ACTIVIDADES FORMATIVAS, METODOLOGÍA A USAR Y REGIMEN DE TUTORIAS

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
--	----

SISTEMAS DE EVALUACIÓN

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,

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

exposiciones en clase, ejercicios, prácticas y trabajo en los talleres a lo largo del curso.

A continuación, se describen cada una de las pruebas y su ponderación:

- Primer ejercicio de evaluación continua (temas 1, 2 y 3) (SE2): 15%.
- Segundo ejercicio de evaluación continua (temas 3, 4 y 5) (SE2): 15%.
- Pruebas de Laboratorio (SE2): Dos pruebas: Total 20%.
- Examen final (todos los temas) (SE1): 50%. Nota mínima en el examen final: 2 (sobre 5)

La nota final de la asignatura se obtiene sumando la nota de las distintas pruebas de evaluación continua. Para superar la asignatura es necesario obtener una nota final igual o superior a 50.

Si un estudiante decide no seguir la evaluación continua (renuncia a las notas obtenidas en las pruebas de evaluación continua), tendrá derecho a realizar un examen final (misma fecha y lugar que el examen ordinario). La nota obtenida en dicho examen equivale al 60% de la nota final (es decir, es necesario obtener un 8.3 sobre 10 para poder superar la asignatura).

Convocatoria Extraordinaria. En esta convocatoria, siempre se considerará la opción más ventajosa para el estudiante: o bien el 100% de la nota del examen extraordinario, o bien el 50% de la nota y el resto de las notas de la evaluación continua.

En las distintas pruebas de evaluación, una solución (por ejemplo, la implementaciones de una estructuras de datos y sus algoritmos) será correcta sólo si cumple los siguientes criterios:

- Cumple las especificaciones descritas en el enunciado.
- Su implementación es correcta y robusta. Es decir, además de no contener errores de sintaxis, para cada posible entrada que recibe, produce la salida correcta.
- La solución debe ser lo más eficiente posible, tanto en términos de complejidad temporal como espacial.
- La solución debe ser clara, limpia, fácil de entender y de mantener.
- El código debe cumplir las recomendaciones definidas en El Zen de Python (<https://peps.python.org/pep-0020/>). El código debe estar correctamente refactorizado.

BIBLIOGRAFÍA BÁSICA

- Aho, A. V.; Hopcroft, J. E.; Ullman, J. D. Estructuras de Datos y Algoritmos, Addison Wesley Iberoamericana.
- Michael T. Goodrich and Roberto Tamassia Data Structures and Algorithms in Python, , John Wiley & Sons, 2013

BIBLIOGRAFÍA COMPLEMENTARIA

- Isabel Segura Bedmar, Harith AlJumaily, Julian Moreno Schneider, Juan Perea & Nathan D. Ryan Algorithms and Data Structures, OCW-UC3M: <http://ocw.uc3m.es/ingenieria-informatica/algorithms-and-data-structures>, 2011
- Lourdes Araujo Serna, Raquel Martínez Unanue y Miguel Rodríguez Artacho Programación y estructuras de datos avanzadas, UNED, 2011

RECURSOS ELECTRÓNICOS BÁSICOS

- Isabel Segura Bedmar, Lourdes Moreno, Harith AlJumaily, José Luis Martínez . ESTRUCTURA DE DATOS Y ALGORITMOS: <http://ocw.uc3m.es/ingenieria-informatica/estructura-datos-algoritmos>