Universidad Zaragoza

Curso Académico: 2022/23

30213 - Estructuras de datos y algoritmos

Información del Plan Docente

Año académico: 2022/23
Asignatura: 30213 - Estructuras de datos y algoritmos
Centro académico: 110 - Escuela de Ingeniería y Arquitectura
326 - Escuela Universitaria Politécnica de Teruel
Titulación: 439 - Graduado en Ingeniería Informática
443 - Graduado en Ingeniería Informática
Créditos: 6.0
Curso: 2
Periodo de impartición: Primer semestre
Clase de asignatura: Obligatoria
Materia:

1. Información Básica

1.1. Objetivos de la asignatura

La asignatura y sus resultados previstos responden a los siguientes planteamientos y objetivos:

En esta asignatura el alumno mejorará su capacidad para diseñar y desarrollar programas de ordenador haciendo énfasis en la identificación, diseño y definición de Tipos Abstractos de Datos (TADs) independientemente de su implementación. El alumno aprenderá a diseñar e implementar TADs para que sean reutilizables, eficientes y robustos, y a implementarlos garantizando dichas propiedades. Se presentarán algunos de los TADs fundamentales de uso más frecuente, como: pilas, colas, listas, árboles de búsqueda, tablas, etc., para los que se estudiarán y compararán distintas alternativas de implementación. También se introducirán una serie de esquemas algorítmicos básicos (como dividir para vencer, búsqueda con retroceso, voracidad...) y el alumno aprenderá a reconocer los problemas que requieren este tipo de esquemas para su resolución y cómo aplicarlos.

Estos planteamientos y objetivos están alineados con algunos de los Objetivos de Desarrollo Sostenible, ODS, de la Agenda 2030 (https://www.un.org/sustainabledevelopment/es/) y determinadas metas concretas, de tal manera que la adquisición de los resultados de aprendizaje de la asignatura proporciona capacitación y competencia al estudiante para contribuir en cierta medida a su logro:

- Objetivo 8: Promover el crecimiento económico sostenido, inclusivo y sostenible, el empleo pleno y productivo y el trabajo decente para todo.
  Meta 8.4. Mejorar progresivamente, de aquí a 2030, la producción y el consumo eficientes de los recursos mundiales y procurar desvincular el crecimiento económico de la degradación del medio ambiente, conforme al Marco Decenal de Programas sobre modalidades de Consumo y Producción Sostenibles, empezando por los países desarrollados.
- Objetivo 16: Promover sociedades, justas, pacíficas e inclusivas.
  Meta 16.5. Reducir considerablemente la corrupción y el soborno en todas sus formas.

1.2. Contexto y sentido de la asignatura en la titulación

Estructuras de Datos y Algoritmos (EDA) es una asignatura obligatoria englobada en la materia de formación común en Programación y Computación.

Esta asignatura completa la formación recibida por el alumno en las asignaturas de Programación I y Programación II, y le prepara para abordar proyectos de programación de cada vez mayor tamaño y complejidad, y a hacerlo aplicando mejores técnicas y estrategias en el diseño e implementación, permitiendo además el reparto efectivo de la carga del trabajo de implementación y desarrollo de las diferentes partes del sistema a desarrollar. Esta línea de formación continuará ampliándose en las asignaturas de Tecnología de Programación, Programación de Sistemas Concurrentes y Distribuidos, e Ingeniería de Software, así como en otras asignaturas posteriores en el plan de estudios

Además, algunos de los TADs y algoritmos estudiados en EDA serán necesarios para diversas asignaturas que tienen a EDA como prerrequisito directo o indirecto, tales como Tecnología de Programación, Bases de Datos, e Inteligencia Artificial.

1.3. Recomendaciones para cursar la asignatura

El alumno que curse esta asignatura ha de contar con una formación en programación del nivel correspondiente al necesario para superar la asignatura de Programación II. Por otra parte una adecuada formación matemática del nivel de la asignatura Matemática Discreta resulta muy conveniente.

2. Competencias y resultados de aprendizaje

2.1. Competencias

Al superar la asignatura, el estudiante será más competente para...

Reconocer y aplicar los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

Resolver problemas reconociendo o diseñando, y utilizando de forma eficiente, los tipos y estructuras de datos más adecuados a la resolución de un problema.

Resolver problemas y tomar decisiones con iniciativa, creatividad y razonamiento crítico.

Aprender de forma continuada y desarrollar estrategias de aprendizaje autónomo.

2.2. Resultados de aprendizaje

El estudiante, para superar esta asignatura, deberá demostrar los siguientes resultados...

Es capaz de identificar, diseñar y definir Tipos Abstractos de Datos (TADs) independientemente de su implementación.

Diseña e implementa TADs reutilizables y robustos en un lenguaje de programación modular o en un lenguaje orientado a objetos.

Diseña e implementa programas robustos de tamaño medio identificando, definiendo e implementando los Tipos Abstractos de Datos (TADs) necesarios.

Es capaz de identificar, utilizar e implementar algunos TADs fundamentales, como: pilas, colas, listas, árboles de búsqueda, tablas hash y grafos.

Es capaz de comparar distintas alternativas de implementación de TADs con respecto al tiempo de ejecución de algoritmos y al uso de la memoria, y de seleccionar la más adecuada en cada problema o contexto.

Conoce y aplica los esquemas algorítmicos básicos (como dividir para vencer, búsqueda con retroceso, voracidad...) a la resolución de problemas.

2.3. Importancia de los resultados de aprendizaje

Estructuras de Datos y Algoritmos constituirá una base sólida en la formación del alumno para el diseño y desarrollo de sistemas o proyectos de programación de cada vez mayor tamaño y complejidad, ya sean basados en el diseño modular o en el diseño orientado a objetos, buscando siempre la encapsulación, calidad, eficiencia y reutilización del software.

Se presentarán además un conjunto de TADs y algoritmos de uso frecuente, y que todo futuro Ingeniero Informático debe conocer y saber utilizar para poder diseñar soluciones en los nuevos contextos o problemas a los que se enfrente.

3. Evaluación

3.1. Tipo de pruebas y su valor sobre la nota final y criterios de evaluación para cada prueba

La prueba global de evaluación de la asignatura consta de dos partes:

1. **Prácticas y trabajos de laboratorio.** Se propondrán trabajos de programación en equipo cuyo resultado se plasmará en la entrega de los programas resultantes convenientemente diseñados y documentados. El trabajo presentado por cada equipo de estudiantes se calificará con una nota cuantitativa de 0 a 10. Para obtener dicha nota se valorará el funcionamiento del programa según las especificaciones, la calidad de su diseño y su presentación y la adecuada aplicación de los métodos de resolución. Es necesaria una calificación mínima de 5.0 puntos en esta parte para aprobar la asignatura. En tal caso la calificación obtenida pondera un 20% de la nota final de la asignatura.
2. **Examen escrito** en el que se deberán resolver problemas de programación y, en su caso, responder preguntas conceptuales o resolver algún ejercicio. Se calificará con una nota de 0 a 10. En general, se valorará la calidad y claridad de las respuestas y soluciones propuestas, su adecuación a las especificaciones y restricciones planteadas, la calidad del diseño, la adecuada aplicación de los métodos de resolución y el tiempo empleado. Será necesario obtener una calificación mínima de 5.0 puntos en el examen escrito para aprobar la asignatura. En tal caso la calificación obtenida pondera un 80% de la nota final de la asignatura.

Si la calificación obtenida por el alumno en el examen escrito es igual o superior a 5.0 y su calificación en los trabajos de

laboratorio es igual o superior a 5.0, entonces la calificación del alumno en la asignatura se obtendrá como la suma ponderada de las calificaciones del examen escrito con ponderación del 80% y del trabajo de programación con ponderación del 20%. Si ambas calificaciones (examen escrito y trabajos de laboratorio) son menores que 5.0, la calificación en la asignatura se obtendrá de igual forma (ponderando un 80% y un 20%, respectivamente). Si, por el contrario, la calificación obtenida por el alumno en una de las dos pruebas (examen escrito o trabajo de laboratorio) es menor que 5.0, entonces la calificación del alumno en la asignatura será igual a esa nota que no alcanza el 5.0.

Segunda convocatoria. La evaluación de la asignatura se realiza en base a dos partes análogas a las de la primera convocatoria, con las mismas ponderaciones y exigencia de notas mínimas.

4. Metodología, actividades de aprendizaje, programa y recursos

4.1. Presentación metodológica general

El proceso de aprendizaje que se ha diseñado para esta asignatura se basa en lo siguiente:

1. El estudio y trabajo continuado desde el primer día de clase.
2. El aprendizaje de conceptos y metodologías para el diseño e implementación de TADs correctos, reutilizables y eficientes a través de las clases magistrales, en las que se favorecerá la participación de los alumnos.
3. La aplicación de tales conocimientos al diseño y análisis de algoritmos y programas en las clases de problemas. En estas clases los alumnos desempeñarán un papel activo en la discusión y resolución de los problemas.
4. Las clases de prácticas en laboratorio en las que el alumno deberá poner en práctica la tecnología necesaria para desarrollar proyectos de programación de tamaño pequeño o medio, utilizando un lenguaje de programación determinado y aplicando los conceptos y técnicas estudiadas en esta asignatura.
5. El trabajo en equipo desarrollado para resolver un problema propuesto basado en lo hecho en las prácticas y cuyo resultado se plasma en la entrega del programa resultante convenientemente diseñado y documentado.

En definitiva, una labor continuada en la que se conjugue la comprensión de conceptos, el análisis y la resolución de problemas de programación utilizando "lápiz y papel" y la puesta a punto en computador de proyectos de programación de tamaño pequeño o medio.

4.2. Actividades de aprendizaje

El programa que se ofrece al estudiante para ayudarle a lograr los resultados previstos comprende las siguientes actividades:

- En las clases impartidas en el aula se desarrollará el temario de la asignatura.
- En las clases de problemas se resolverán problemas de aplicación de los conceptos y técnicas presentadas en el programa de la asignatura. Se propondrán problemas y ejercicios para ser resueltos antes de la clase de problemas en la que se presentarán y discutirán diferentes soluciones a dichos problemas. También se propondrán ejercicios durante la sesión de problemas para ser resueltos durante la misma, algunos de forma individual y otros para ser trabajados en grupo.
- Las sesiones de prácticas se desarrollan en un laboratorio informático. En estas sesiones el alumno deberá trabajar en equipo y realizar una serie de trabajos de programación directamente relacionados con los temas estudiados en la asignatura. Para las prácticas de laboratorio se propondrán una serie de trabajos o ejercicios de programación para que el alumno los resuelva, desarrolle parte del trabajo en el laboratorio, los complete como trabajo en casa, y los entregue.

4.3. Programa

1. Programación con Tipos Abstractos de Datos.
2. Tipos de datos lineales.
3. Tipos de datos arborescentes.
4. Tipos de datos funcionales.
5. Introducción a los grafos.
6. Introducción a los esquemas algorítmicos.

4.4. Planificación de las actividades de aprendizaje y calendario de fechas clave

En Zaragoza:

La organización docente prevista de la asignatura es la siguiente:

- Clases teóricas (2 horas semanales)
- Clases de problemas (1 hora semanal)
- Clases prácticas (2 horas cada dos semanas)

Trabajo del estudiante

La dedicación del estudiante para alcanzar los resultados de aprendizaje en esta asignatura se estima en 150 horas distribuidas del siguiente modo:

- 30 horas de clases teóricas
- 15 horas de clases de problemas
- 15 horas de clases de prácticas
- 24 horas de trabajo de programación para completar las tareas prácticas
- 60 horas de estudio personal efectivo (estudio de apuntes y textos, resolución de problemas, preparación de clases y prácticas)
- 6 horas de exámenes

El calendario de exámenes y las fechas de entrega de trabajos se anunciará con suficiente antelación.

En Teruel:

La organización docente prevista de la asignatura es la siguiente:

- Clases teóricas (2 horas semanales)
- Clases de problemas (1 hora semanal + 5 horas repartidas a lo largo del cuatrimestre)
- Clases prácticas (2 horas cada dos semanas hasta un máximo de 5 sesiones por grupo)

Trabajo del estudiante

La dedicación del estudiante para alcanzar los resultados de aprendizaje en esta asignatura se estima en 150 horas distribuidas del siguiente modo:

- 30 horas de clases teóricas
- 20 horas de clases de problemas
- 10 horas de clases de prácticas
- 24 horas de trabajo de programación para completar las tareas prácticas
- 60 horas de estudio personal efectivo (estudio de apuntes y textos, resolución de problemas, preparación de clases y prácticas)
- 6 horas de exámenes

El calendario de exámenes y las fechas de entrega de trabajos se anunciará con suficiente antelación.

4.5. Bibliografía y recursos recomendados

http://psfunizar10.unizar.es/br13/egAsignaturas.php?codigo=30213