

Guía docente de la asignatura

## Estructura de Datos (2961122)



Fecha de aprobación: 23/06/2023

<b>Grado</b>	Grado en Ingeniería Informática	<b>Rama</b>	Ingeniería y Arquitectura				
<b>Módulo</b>	Formación Específica de Rama	<b>Materia</b>	Programación e Ingeniería del Software				
<b>Curso</b>	2º	<b>Semestre</b>	1º	<b>Créditos</b>	6	<b>Tipo</b>	Obligatoria

### PRERREQUISITOS Y/O RECOMENDACIONES

Se recomienda la superación de los contenidos y adquisición de competencias de las materias de formación básica relacionadas con la programación, especialmente de Fundamentos de Programación y de Metodología de la Programación.

### BREVE DESCRIPCIÓN DE CONTENIDOS (Según memoria de verificación del Grado)

- Introducción a la eficiencia de algoritmos.
- Tipo de Dato Abstracto (TDA).
- Especificación e implementación de Tipos de Datos Abstractos (TDAs): listas, pilas, colas, árboles, tablas hash, grafos.

### COMPETENCIAS ASOCIADAS A MATERIA/ASIGNATURA

#### COMPETENCIAS GENERALES

- CG08 - Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
- CG09 - Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

#### COMPETENCIAS ESPECÍFICAS

- CE13 - Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

#### COMPETENCIAS TRANSVERSALES



- CT02 - Capacidad para tomar decisiones basadas en criterios objetivos (datos experimentales, científicos o de simulación disponibles) así como capacidad de argumentar y justificar lógicamente dichas decisiones, sabiendo aceptar otros puntos de vista.
- CT05 - Capacidad de trabajo en equipo, usando competencias demostrables mediante la elaboración y defensa de argumentos.

## RESULTADOS DE APRENDIZAJE (Objetivos)

- Reconocer la importancia de la abstracción y conocer los tipos de abstracciones que aparecen en programación: funcional, de datos, de iteradores y abstracción por generalización.
- Saber diferenciar entre la especificación, representación e implementación de un tipo de dato abstracto, conociendo los conceptos de Función de Abstracción e Invariante de la Representación.
- Comprender cómo los conceptos de ocultamiento de información y encapsulamiento ayudan al desarrollo de tipos de datos más fiables.
- Comprender los métodos de especificación: basados en una definición mediante axiomas o el método constructivo u operacional (basado en el uso de precondiciones y postcondiciones).
- Ser capaz de diseñar e implementar pequeñas aplicaciones para cada uno de los distintos tipos de datos que se imparten en la materia (por ejemplo, listas, pilas, colas, colas con prioridad, conjuntos, diccionarios, árboles, tablas hash, grafos).
- Adquirir la capacidad para comprender cómo el uso de distintos tipos de datos afecta a la eficiencia de los algoritmos que la usan.
- Ser capaz de implementar en lenguajes de alto nivel los tipos de datos propios de la materia así como otros definidos por el usuario.
- Conocer las distintas representaciones e implementaciones de los tipos de datos que se imparten en la materia.
- Ser capaz de comparar implementaciones alternativas para un tipo de dato analizando los factores que influyen en la eficiencia y el uso de memoria.
- Adquirir la capacidad de evaluar las necesidades de una aplicación específica, tomando decisiones justificadas sobre los tipos de datos y la representación más adecuadas.

## PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

### TEÓRICO

- Tema 1. Introducción a la eficiencia de los algoritmos.
- Tema 2. Abstracción de datos
- Tema 3. Tipos de datos contenedores básicos.
  - Pilas, colas, colas con prioridad.
  - Conjuntos, diccionarios.
  - Vectores y Listas.
  - Implementación de TDAs básicos.
- Tema 4. Tipos de datos contenedores complejos.
  - Árboles, tablas hash, grafos
  - Implementación de TDAs complejos.

### PRÁCTICO



- Eficiencia de algoritmos. Con diversos ejemplos analizar la eficiencia teórica vs. eficiencia empírica.
- Construcción de TDAs básicos.
- Uso e Implementación de TDAs lineales.
- Uso e Implementación de TDAs no lineales.
- Seminarios:
  - Aplicación de TDA sobre problemas reales.
  - Uso de la STL en problemas prácticos.

## BIBLIOGRAFÍA

### BIBLIOGRAFÍA FUNDAMENTAL

- Rosa Rodríguez-Sánchez, J. Fdez-Valdivia, J.A. García, Estructuras de Datos en C++. Un enfoque práctico.. 2020
- A. Garrido, J. Fdez-Valdivia, Abstracción y Estructuras de Datos en C++. Delta public.. 2006
- A. Garrido. Estructuras de Datos avanzadas con soluciones en C++. Editorial Universidad de Granada 2018.
- A. Garrido, Programación genérica en C++: L a biblioteca estándar. Editorial Universidad de Granada 2017.
- A. Garrido. Fundamentos de Programación con la STL. Editorial Universidad de Granada, 2016
- R. Musser, J. Derge y A. Saini. STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library. 3 Edición. Addison-Wesley 2009.
- Robson, R. Using the STL (2nd ed.) Springer Verlag. 2013
- Frank M. Carrano. Data Abstraction & Problem Solving with C++: Walls and Mirrors. 7ed. Pearson. 2017
- Elliot B. Koffman and Paul AT Wolfgang. Objects, Abstraction, Data Structures and Design: Using C++. Wiley. 2006

### BIBLIOGRAFÍA COMPLEMENTARIA

- A. Garrido. Metodología de Programación: de bits a objetos. Editorial Universidad de Granada, 2016
- H.Deitel and P. Deitel. C++ How to program (Early Objects Version) 9/E (2013) Pearson
- Gilberg, R.F., Forouzan, B.A. . Data structures: A pseudocode approach with C++. Brooks/Cole. 2001
- N.M. Josuttis, The C++ Standard Library: A Tutorial and Reference. Addison-Wesley. 2012.
- B. Liskov. J. Guttag. Program Development in Java: Abstraction, Specification, and Object- Oriented Design. Addison-Wesley. 2000.
- Bjarne Stroustrup. "The C++ Programming Language, 4th Edition". Addison Wesley Professional, 2013.
- Dale, Nell. C++ Plus Data Structures. Jones & Bartlett Learning; 6a edición. 2016
- Varsha H. Patil . Data Structures using C++ . New Delhi, India: Oxford University Press, .2018

## ENLACES RECOMENDADOS



- [C++ Reference](#) (en inglés)
- [C Plus Plus](#) (en inglés)

## METODOLOGÍA DOCENTE

- MD01 - Lección Magistral (Clases Teóricas-Expositivas)
- MD02 - Actividades Prácticas (Resolución de Problemas, Resolución de Casos Prácticos, Desarrollo de Proyectos, Prácticas en Laboratorio, Taller de Programación, Aula de Informática, Prácticas de Campo).
- MD03 - Seminarios (Debates, Demos, Exposición de Trabajos Tutelados, Conferencias, Visitas Guiadas, Monografías).
- MD04 - Actividades no presenciales Individuales.
- MD05 - Actividades no presenciales Grupales.
- MD06 - Tutorías Académicas.

## EVALUACIÓN (instrumentos de evaluación, criterios de evaluación y porcentaje sobre la calificación final)

### EVALUACIÓN ORDINARIA

Porcentajes de evaluación	Ponderación
Actividades Formativas	
Parte teórica	60%
Parte práctica	30%
Otros (seminarios, participación, etc)	10%

Se utilizará el siguiente sistema de evaluación:

- Para la parte teórica se realizará un examen final. La ponderación de este bloque es del 60%
- Para la parte práctica se realizarán prácticas de laboratorio, resolución de problemas, desarrollo de proyectos (individuales o en grupo), y/o eventuales entregas de ejercicios sobre el desarrollo y los resultados de las actividades propuestas. La ponderación de este bloque es del 30%
- En su caso, la parte de trabajo autónomo y los seminarios se evaluarán teniendo en cuenta la asistencia a los seminarios, los problemas propuestos que hayan sido resueltos y entregados por los estudiantes o la presentación oral de los trabajos desarrollados. La ponderación de estos es del 10%

La calificación global será una calificación numérica obtenida mediante la suma de las anteriores calificaciones.

### EVALUACIÓN EXTRAORDINARIA

Examen final con preguntas teóricas y prácticas con un valor de 10 puntos. El estudiante podrá guardar la nota obtenida en la convocatoria de Enero en los bloques de "Parte Práctica" y "Otros" y en ese caso la calificación final del examen se ajustará a 6 puntos.

### EVALUACIÓN ÚNICA FINAL

De acuerdo a lo establecido en la Normativa de evaluación y de calificación de los estudiantes de la Universidad de Granada aprobada en Consejo de Gobierno de 20 de mayo de 2013 (NCG71/2), la



evaluación será preferentemente continua. No obstante, el estudiante que no pueda acogerse a dicho sistema por motivos laborales, estado de salud, discapacidad o cualquier otra causa debidamente justificada podrá acogerse a la evaluación única final. Para ello deberá solicitarlo al Director del Departamento en las dos primeras semanas de impartición de la asignatura o, excepcionalmente, en las dos primeras semanas tras la matriculación en la asignatura (NCG78/9: Instrucción relativa a la aplicación del artículo 8.2).

Esta modalidad de evaluación se realizará en un único acto académico en la fecha establecida por el Centro y consistirá en un examen escrito (evaluado de 0 a 10) que incluirá preguntas tanto de tipo teórico como práctico que garanticen que el estudiante ha adquirido la totalidad de las competencias descritas en esta misma guía docente.

Todo lo relativo a la evaluación se regirá por la normativa sobre planificación docente y organización de exámenes vigente en la Universidad de Granada.

El sistema de calificaciones se expresará mediante calificación numérica de acuerdo con lo establecido en el art. 5 del R. D 1125/2003, de 5 de septiembre, por el que se establece el sistema europeo de créditos y el sistema de calificaciones en las titulaciones universitarias de carácter oficial y validez en el territorio nacional.

### INFORMACIÓN ADICIONAL

- Se recomienda la asistencia regular a las clases teóricas y prácticas.
- Plataforma docente usada: [PRADO](#)

