

# Estructuras de Datos y Algoritmos

## Presentación de la asignatura

© All wrongs reversed – bajo licencia CC-BY-NC-SA 4.0



1542

**Universidad**  
Zaragoza

Dpto. de Informática e Ingeniería de Sistemas  
Universidad de Zaragoza, España

Curso 2024/2025

**Grado en Ingeniería Informática**

UNIVERSIDAD DE ZARAGOZA

*Aula 0.04, Edificio Agustín de Betancourt*



# Índice

- 1 Profesorado
- 2 Descripción de la asignatura
- 3 Evaluación
- 4 Carga de trabajo para el estudiante
- 5 Bibliografía
- 6 Tareas

# Índice

- 1 Profesorado
- 2 Descripción de la asignatura
- 3 Evaluación
- 4 Carga de trabajo para el estudiante
- 5 Bibliografía
- 6 Tareas

# Profesorado

## Coordinador de la asignatura

- Ricardo J. Rodríguez ([rjrodriguez@unizar.es](mailto:rjrodriguez@unizar.es))

## Profesores de teoría y problemas



Ricardo J. Rodríguez (mañanas)  
[rjrodriguez@unizar.es](mailto:rjrodriguez@unizar.es)



Yolanda Villate (tardes)  
[yvillate@unizar.es](mailto:yvillate@unizar.es)

# Profesorado

## Información de interés para grupo de mañanas

### Ricardo J. Rodríguez

- Despacho D0.05
- **Tutorías:** M de 12:00 a 14:00, X de 17:00 a 19:00 y V de 14:00 a 16:00
  - Reservar con un mínimo de 24 horas de antelación
  - Posibles cambios puntuales (mirar el calendario de citas)
  - Reservas a través de [Google Calendar](#) (enlace también disponible en Moodle)
    - Indicar en la reserva: *signatura, tema a tratar, presencial/remoto*

<https://webdiis.unizar.es/asignaturas/EDA/>

# Profesorado

## Profesores de prácticas



Jordi Bernad  
jbernad@unizar.es



Javier Carrillo-Mondéjar  
jcarrillo@unizar.es



Yolanda Villate

# Índice

- 1 Profesorado
- 2 Descripción de la asignatura**
- 3 Evaluación
- 4 Carga de trabajo para el estudiante
- 5 Bibliografía
- 6 Tareas

# Descripción de la asignatura

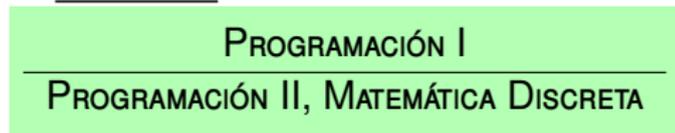
## ESTRUCTURAS DE DATOS Y ALGORITMOS

- Asignatura *obligatoria* de 6 créditos ECTS
- Formación previa en programación (necesario) y matemática (recomendable)
  - PROGRAMACIÓN II
  - MATEMÁTICA DISCRETA
- **Objetivo principal**
  - Identificar, diseñar y definir Tipos Abstractos de Datos, independientemente de su implementación
- Enlace a curso en Moodle:  
<https://moodle.unizar.es/add/course/view?id=106556>
  - Clave de auto-matrícula: \*\*\*\*\*

# Descripción de la asignatura

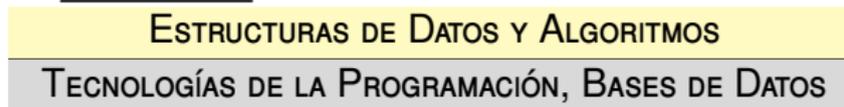
## Relación con otras asignaturas de la carrera

Primer curso

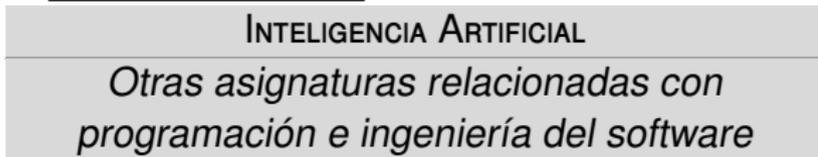


- Diseño imperativo descendente (modularización)
- Diseño recursivo
- Estructuras de datos sencillas (vectores, ficheros sec.)
- Análisis de la eficiencia de algoritmos (caso peor)
- Grafos (conceptos, representación y algoritmos básicos)

Segundo curso



Tercer curso y siguientes



# Descripción de la asignatura

## Resultados de aprendizaje

- Definir tipos abstractos de datos (TAD's) independientemente de su implementación.
- Implementar TAD's en un lenguaje de programación modular.
- Implementar y utilizar algunos TAD's fundamentales, como pilas, colas, listas, árboles de búsqueda, tablas hash y grafos.
- Comparar distintas alternativas de implementación de TAD's con respecto al tiempo de ejecución de algoritmos y al uso de la memoria.
- Afrontar el diseño modular de programas de tamaño medio identificando, definiendo e implementando los TAD's necesarios.
- *Aplicar los esquemas algorítmicos básicos (como dividir para vencer, búsqueda con retroceso, voracidad...) a la resolución de problemas.*

Más información: [Guía docente de la asignatura, curso 2024/2025](#)

# Descripción de la asignatura

## Programa

**T1** Programación con Tipos Abstractos de Datos

**T2** Tipos de datos lineales

**T3** Tipos de datos arborescentes

**T4** Tipos de datos funcionales

**T5** Introducción a los grafos

**T6** Introducción a los esquemas algorítmicos

# Descripción de la asignatura

## Metodología

### Clases de teoría

- Aprendizaje de conceptos
- 2 horas semanales
- Asistencia voluntaria (**pero muy recomendable**)

### Clases de problemas

- Aplicación de conceptos
- 1 hora semanal
- Grupos desdoblados
- Asistencia voluntaria (**pero muy recomendable**)

# Descripción de la asignatura

## Metodología

### Clases prácticas

- 5 grupos de prácticas
- 2 horas en semanas alternas

# Descripción de la asignatura

## Prácticas

### **5 prácticas, distribuidas en 6 sesiones:**

- P1** Implementación de un TAD en memoria estática
- P2** Implementación de un TAD genérico en memoria estática
- P3** Implementación de TADs lineales en memoria dinámica (dos sesiones)
- P4** Implementación de TADs arborescentes en memoria dinámica
- P5** Diseño de programas basado en TADs

# Descripción de la asignatura

## Prácticas

- **Elegid grupo de prácticas de los previstos para el grupo de mañanas**
- Reserva plaza en [tarea disponible en Moodle](#)
- Plazo límite: **10 de septiembre (a las 19:00)**
- Horarios:
  - **Lunes** semana **A**, de **08:00 a 10:00**, Laboratorio **L0.01**
  - **Lunes** semana **B**, de **08:00 a 10:00**, Laboratorio **L0.01**
  - **Miércoles** semana **A**, de **12:00 a 14:00**, Laboratorio **L2.11**
  - **Miércoles** semana **B**, de **12:00 a 14:00**, Laboratorio **L2.11**
  - **Jueves** semana **B**, de **12:00 a 14:00**, Laboratorio **L0.01**

# Descripción de la asignatura

## Prácticas

- P1 y P2 son **individuales**
- P3 a P5 por **parejas (del mismo grupo de laboratorio)**
  - **Tenéis que formar esas parejas**
  - Reserva en [tarea disponible en Moodle](#)
  - Plazo límite: **7 de octubre (a las 19:00)**
    - Tras el plazo, **se formarán equipos entre aquellos que no estén inscritos en ninguna pareja**
  - [Foro “Buscando pareja de prácticas” en Moodle](#)
- Cada pareja de prácticas tendrá un profesor tutor
- Lenguaje: C++
- Computador: `hendrix` (mediante IDE o línea de comandos)
  - Puede hacerse en otra máquina, **pero deberá probarse finalmente en `hendrix`, que es donde se evaluará**
  - Más información: documento [“Realización y entrega de prácticas en `hendrix-ssh`”](#)

# Descripción de la asignatura

## Prácticas

### ■ Hay 2 entregas de prácticas

- Se indicará en los enunciados
- **Para su realización se reutilizarán código de prácticas anteriores**

### ■ El código resultante de las prácticas deberá:

- Estar documentado
- Ser legible
- Utilizar diseño descendente
- Ser ejecutable sin errores
- Mostrar un funcionamiento conforme a lo solicitado

# Índice

- 1 Profesorado
- 2 Descripción de la asignatura
- 3 Evaluación**
- 4 Carga de trabajo para el estudiante
- 5 Bibliografía
- 6 Tareas

# Evaluación de las prácticas

- Las entregas de prácticas se compilarán y ejecutarán en el clúster **hendrix**
- Evaluadas de acuerdo a su comportamiento en esa máquina y a las instrucciones específicas dadas en los enunciados
- Cada entrega tendrá un plazo límite y unas instrucciones para la entrega
  - **No se admitirán entregas fuera de plazo**
  - **Ni entregas por ninguna otra vía que la indicada en las instrucciones**
- Cada entrega se calificará con una nota cuantitativa de 0 a 10
- Aspectos a valorar:
  - Funcionamiento de los programas según especificaciones
  - Calidad de su diseño y su presentación
  - Adecuada aplicación de los métodos de resolución
  - Capacidad de los integrantes del equipo para explicar y justificar el diseño realizado, si se les requiere
- **NOVEDAD (excepcionalmente)**
  - *Si la nota final de prácticas en el curso 23-24 es  $\geq 7.0$ , se guarda para el curso 24-25*

# Evaluación global

- Examen escrito (**70 % de la nota final**)
  - Sin apuntes
  - Problemas de programación, preguntas conceptuales o resolver algún ejercicio
  - **Calificación mínima exigida de 5.0**
- Examen práctico de programación, en laboratorio (**30 % de la nota final**)
  - Ejercicios de programación similares a los realizados en prácticas o en clase
  - **Calificación mínima exigida de 5.0**
  - Aquellos que obtengan una nota final mínima de 5.0 en sus entregas de prácticas estarán exentos de realizar este examen práctico
    - Prevalece la nota obtenida en el examen práctico individual

## Convocatorias

- **Primera:** jueves 9 de enero, 2025 (turno M)
- **Segunda:** miércoles 18 de junio, 2025 (turno M)

# Evaluación global

## Plagios

- Plagio total o parcial en las prácticas → **fraude académico**
  - Art. 30 «Prácticas irregulares en Reglamento de Normas de Evaluación del Aprendizaje las pruebas de evaluación distintas de fraude académico»,
  - Art. 31 «Faltas disciplinarias», [Reglamento de Normas de Convivencia](#).

# Evaluación global

```
nota_ponderada := nota_exam_escrito*0.7 + nota_prácticas*0.3
si nota_exam_escrito >= 5.0 and nota_prácticas >= 5.0 entonces
    nota_final := nota_ponderada
sino {suspenseo seguro}
    nota_final := mínimo(4.0, nota_ponderada)
fsi
```

- Calificaciones obtenidas en las dos partes (examen escrito y prácticas) en primera convocatoria **se guardan para la segunda convocatoria del mismo curso académico**

# Índice

- 1 Profesorado
- 2 Descripción de la asignatura
- 3 Evaluación
- 4 Carga de trabajo para el estudiante**
- 5 Bibliografía
- 6 Tareas

# Carga de trabajo para el estudiante

- Asignatura de 6 ECTS → ~ 150 horas (en promedio) efectivas de trabajo
  - **60 horas de actividades presenciales activas**
    - 30 horas de clases de teoría
    - 15 horas de clases de problemas
    - 15 horas de clases prácticas
  - **25 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 clases y prácticas
    - Desarrollo de programas
  - **5 horas de pruebas de evaluación**

# Carga de trabajo para el estudiante

## Recordatorio de tiempos

- Un curso académico son 60 ECTS
- 1 ECTS, en media, son 25 horas de trabajo
  - 1 curso académico equivale a 1500 horas de trabajo
  - 2 meses de vacaciones → 150 horas de trabajo al mes

**Cada semana supone 37.5 horas de trabajo**

# Descripción de la asignatura

## ¿Cómo puedo facilitar mi aprendizaje?

### ■ **Asistir a clase**

- Habiendo realizado el trabajo previo que se haya encargado
- Atendiendo y participando activamente en la clase
- Repasando después y comprendiendo cada lección

### ■ **Resolver los problemas propuestos en las clases de problemas**

### ■ **Resolver y repasar los problemas propuestos en las prácticas**

- Lápiz y papel
- Programación en computador
- Validación del código desarrollado

### ■ **Discutir ideas con otros compañeros** (ojo, no compartir código de prácticas)

### ■ **Consultar dudas a los profesores en sus horarios de tutorías**

# Índice

- 1 Profesorado
- 2 Descripción de la asignatura
- 3 Evaluación
- 4 Carga de trabajo para el estudiante
- 5 Bibliografía**
- 6 Tareas

# Bibliografía

<https://psfunizar10.unizar.es/br13/egAsignaturas.php?id=8027>

## Bibliografía básica

- *Data Structures and Algorithm Analysis in C++*, M.A. Weiss, 4th Edition, Pearson/Addison Wesley, 2013.
- *Fundamentos de Estructuras de Datos. Soluciones en Ada, Java y C++*, Z.J. Hernández y otros, Thomson, 2005.
- *Data Structures and Algorithm Analysis in C++*, C.A. Shaffer, 3rd Edition, Dover Publications, 2011.
- *Apuntes de Estructuras de datos y algoritmos*, J. Campos, 2da ed., v4, 2022.

## C++

- *The C++ Programming Language*, B. Stroustrup, 4th Edition, Addison-Wesley, 2013.

# Bibliografía

<https://psfunizar10.unizar.es/br13/egAsignaturas.php?id=8027>

## Ejercicios

- *Estructura de datos y algoritmos. Ejercicios y problemas resueltos*, N. Martí Oliet, Y. Ortega Mallén, J.A. Verdejo López, Pearson Prentice Hall, 2003.
- *Estructura de datos. Libro de problemas*, L. Joyanes, I. Zahonero, M. Fernández y L. Sánchez, McGraw Hill, 1999.

## Bibliografía complementaria

- *Estructuras de datos. Especificación, diseño e implementación*, X. Franch, 3a edición, Ediciones UPC, 2001.
- *Handbook of Data Structures and Applications*, D.P. Mehta y S. Sahni, Chapman & Hall/CRC, 2005

# Índice

- 1 Profesorado
- 2 Descripción de la asignatura
- 3 Evaluación
- 4 Carga de trabajo para el estudiante
- 5 Bibliografía
- 6 Tareas**

## Para esta semana

### 1 Verifica si tu grupo de teoría es correcto en Moodle

- En Moodle, entra en la asignatura y ve a
- Busca tu nombre en la lista y comprueba tu **Grupo**

### 2 Escoge un grupo de prácticas de los reservados al grupo de mañanas

- Elección abierta 10 de septiembre (a las 19:00)

### 3 Busca una pareja de prácticas

- Usa el foro "Buscando pareja de prácticas" en Moodle si lo necesitas
- Tanto tú como tu pareja de prácticas debéis estar en el mismo grupo

# Tareas

## Para esta semana

### 1 Verifica si tu grupo de teoría es correcto en Moodle

- En Moodle, entra en la asignatura y ve a
- Busca tu nombre en la lista y comprueba tu **Grupo**

### 2 Escoge un grupo de prácticas de los reservados al grupo de mañanas

- Elección abierta 10 de septiembre (a las 19:00)

### 3 Busca una pareja de prácticas

- Usa el foro "Buscando pareja de prácticas" en Moodle si lo necesitas
- Tanto tú como tu pareja de prácticas debéis estar en el mismo grupo

## Para la próxima semana

### 1 Insíbete en un subgrupo (equipo) de prácticas

- Tanto tú como tu pareja de prácticas debéis inscribiros en el mismo subgrupo (equipo)
- Elección abierta 7 de octubre (a las 19:00)

# Estructuras de Datos y Algoritmos

## Presentación de la asignatura

© All wrongs reversed – bajo licencia CC-BY-NC-SA 4.0



1542

**Universidad**  
Zaragoza

Dpto. de Informática e Ingeniería de Sistemas  
Universidad de Zaragoza, España

Curso 2024/2025

**Grado en Ingeniería Informática**

UNIVERSIDAD DE ZARAGOZA

*Aula 0.04, Edificio Agustín de Betancourt*

