

Estructuras de Datos

Manuel Montenegro Montes
Departamento de Sistemas Informáticos y Computación
Facultad de Informática – Universidad Complutense de Madrid

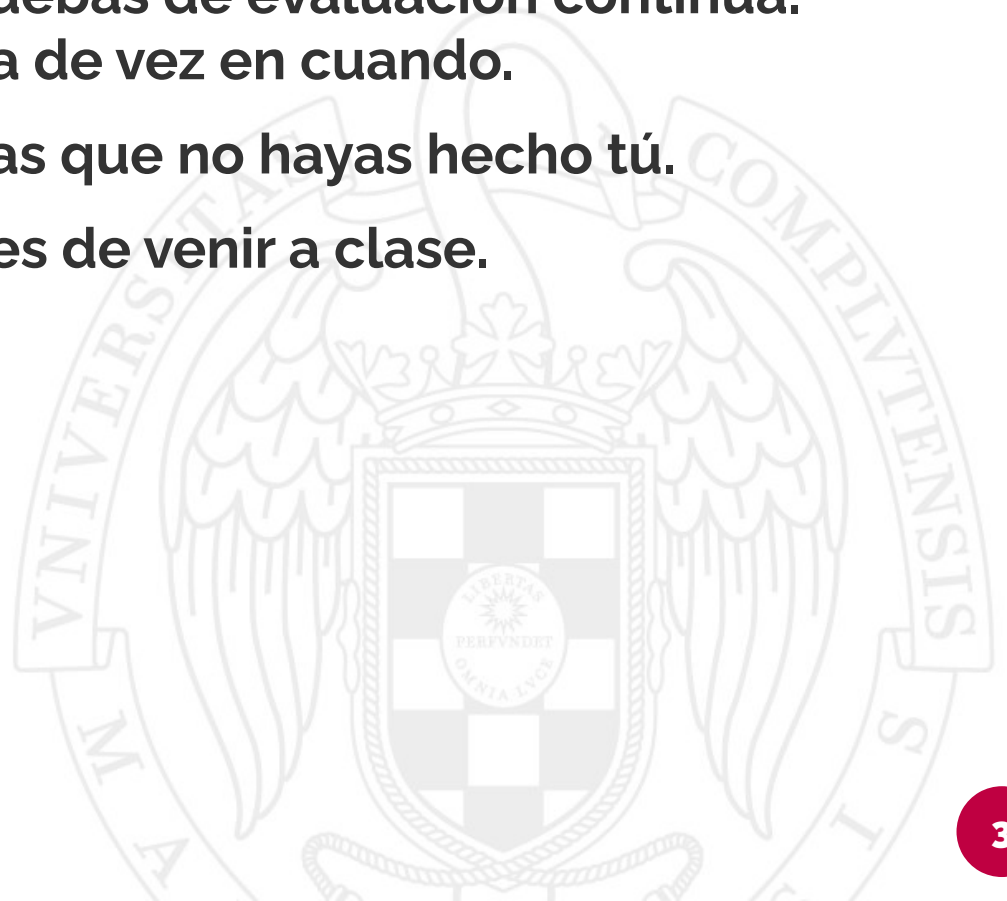


TL;DR

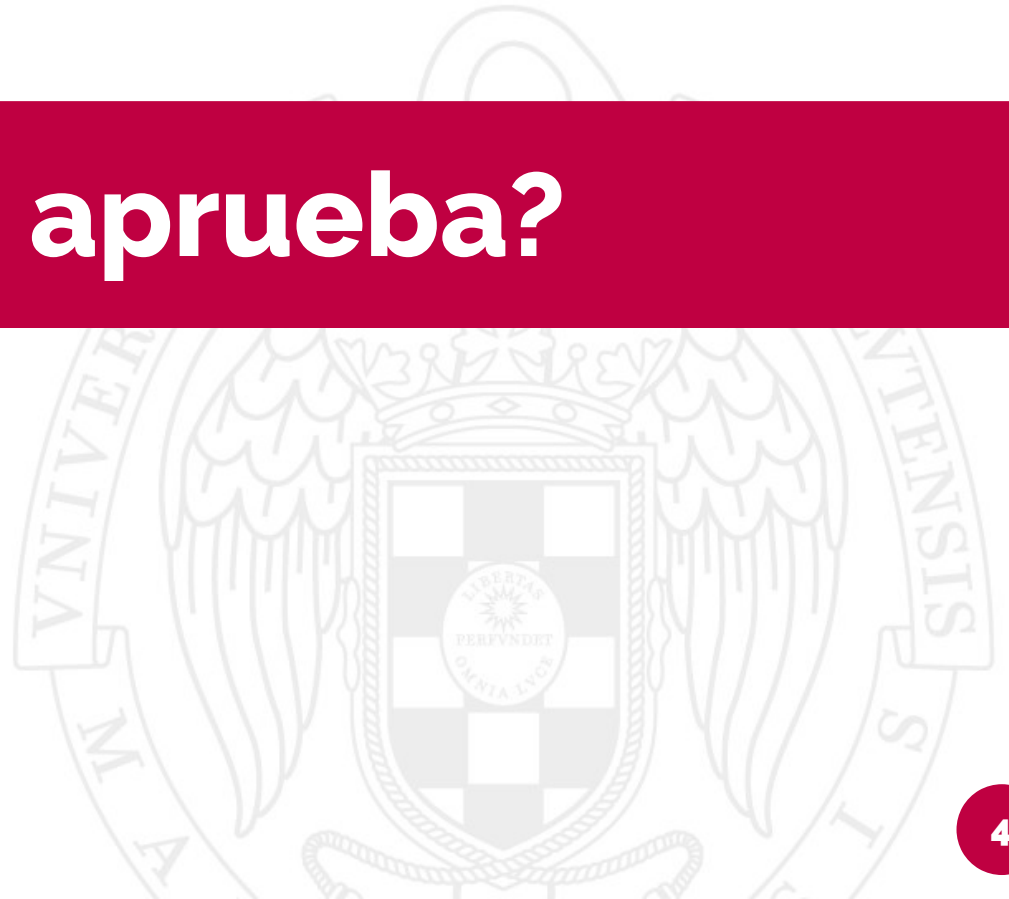


Lo más importante en esta asignatura

- 1) Haz ejercicios.
- 2) En esta asignatura hay muchas pruebas de evaluación continua.
No pasa nada si te «saltas» alguna de vez en cuando.
- 3) No entregues ejercicios ni prácticas que no hayas hecho tú.
- 4) Tendrás que estudiar la teoría antes de venir a clase.




¿Cómo se aprueba?



¿Cómo se aprueba?

Hay dos niveles de habilidad:

- **Conceptual**
 - Conocer y entender el contenido de la materia.
 - Dado un problema, saber cómo resolverlo.
- **Operativa**
 - Resolver problemas **rápido**.
(rapido = el tiempo que dure el examen)
 - Requiere algo de destreza conceptual.



¡En el examen se pide esto!

¿Cómo se consigue la destreza operativa?

Haz ejercicios

Cuanto más, mejor

Haz varios ejercicios de un mismo tipo

¿Cómo se suspende?



Cómo suspender en 3 frases

1) *ED es más fácil que FAL.*

- Sí, la **teoría** es más intuitiva que en FAL.
- Pero el examen consiste en resolver **problemas...**



Cómo suspender en 3 frases

1) *ED es más fácil que FAL.*

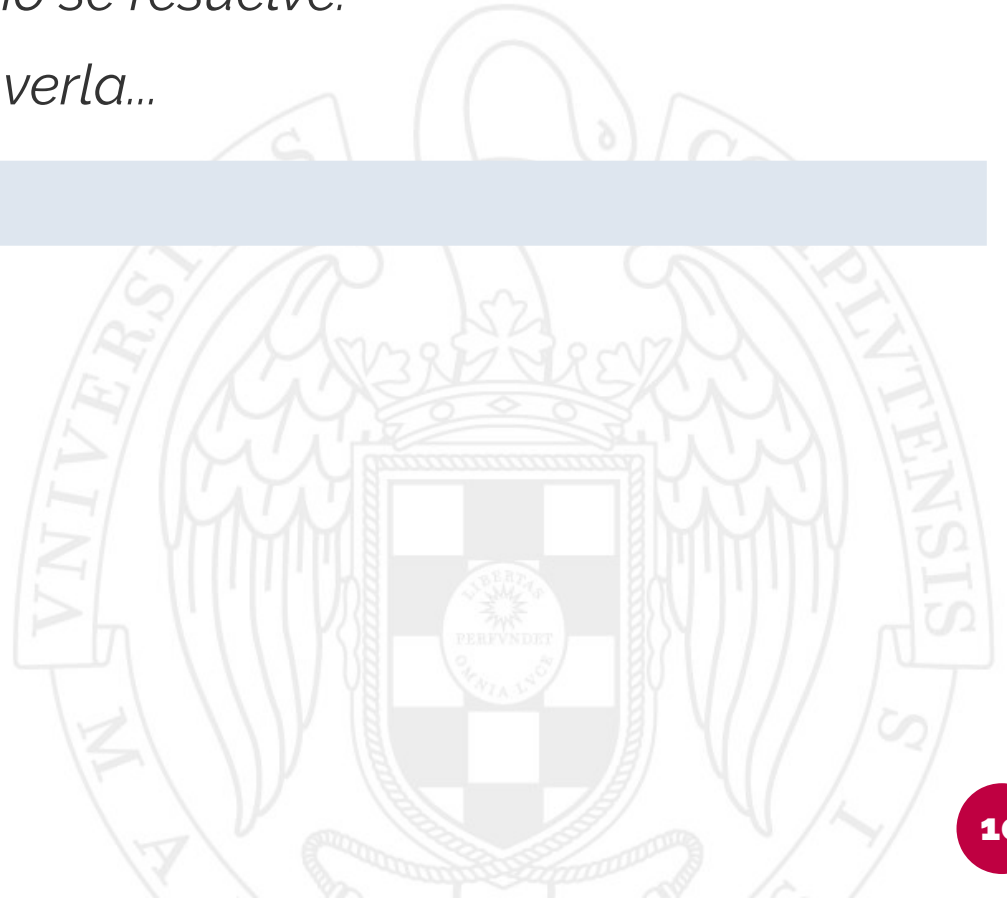
2) *No he hecho el ejercicio, pero sé cómo se resuelve.*

- Resuélvelo y descubrirás cosas que se te han pasado por alto.
- Hay menos ejercicios de “idea feliz” que en FAL, pero son más técnicos.
- No basta con saber hacerlos. Hay que hacerlos rápido.



Cómo suspender en 3 frases

- 1) *ED es más fácil que FAL.*
 - 2) *No he hecho el ejercicio, pero sé cómo se resuelve.*
 - 3) *¿Me pasas la práctica? Es solo para verla...*
- Se masca la tragedia...



Copias, plagios y otros fraudes

- No me da tiempo a entregar un problema. ¿Qué hago?

¡No lo entregues!

- Hay muchas pruebas de evaluación continua.
 - El impacto de cada una en la nota final es de apenas $\approx 0,15$ pts.
 - Además se descartan las calificaciones más bajas antes de hacer la media.

Copias, plagios y otros fraudes

Si entregas una práctica que no has hecho tú

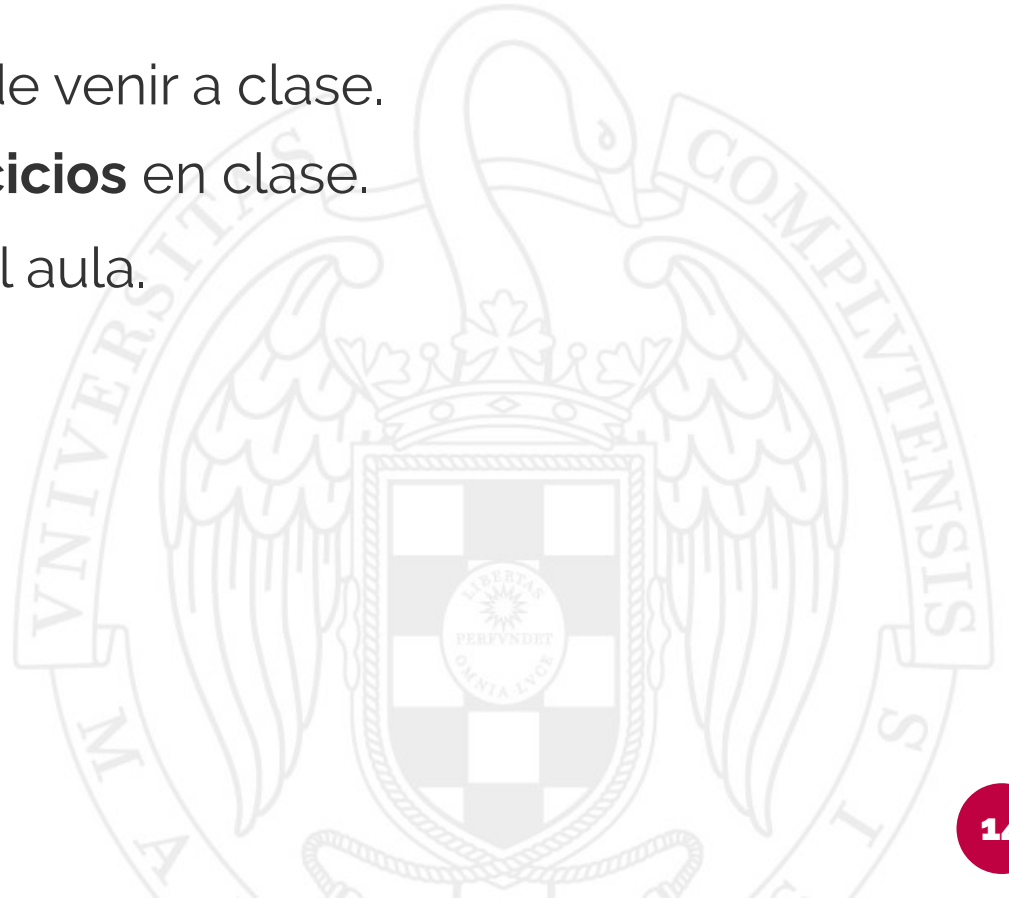
- Calificación de 0 en toda la evaluación continua.
 - Necesitarás $\geq 7,2$ en el examen.
- Notificación a Decanato.
- Se considerará fraude la entrega de prácticas creadas por herramientas de IA generativa (p. ej, ChatGPT).
- Excusas no admitidas:
 - *La entrega la hizo mi compañero/a.*
 - *Hicimos la práctica junto con otro grupo.*

¿Qué haremos en clase?



Funcionamiento de la asignatura

- Se utiliza metodología de clase invertida (*flipped learning*)
- **No se impartirán clases de teoría.**
 - Debéis estudiar la teoría antes de venir a clase.
- Realizaremos exclusivamente **ejercicios** en clase.
 - Tanto en laboratorio, como en el aula.



[1] Teoría

- Estudio individual del material que se colgará en el Campus Virtual.
- El material consiste en:
 - Vídeos (elaborados por el profesor).
 - Cuestionarios de autoevaluación.
 - Problemas propuestos de autoevaluación.
- Las actividades de autoevaluación no forman parte de la evaluación continua, pero son útiles para el estudio.
- El material se colgará el jueves de cada semana, y tendrá que estudiarse antes del jueves de la semana siguiente.

[2] Clases en aula

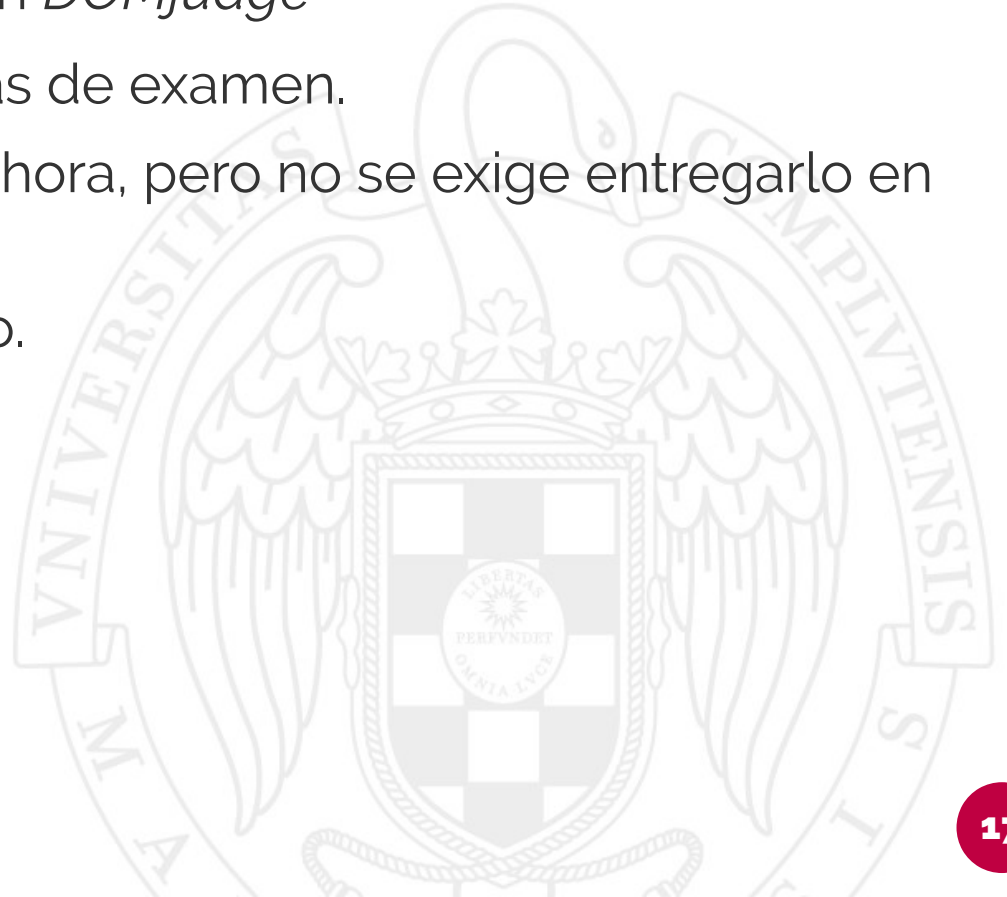
Jueves y viernes de cada semana (12-13h).

- Cuestionario de control al principio de la clase del jueves.
 - Dificultad similar a los de autoevaluación.
 - Las respuestas incorrectas no restan puntos.
 - Individual, en la mayoría de los casos.
 - La calificación forma parte de la **evaluación continua**.
- Resolución de problemas:
 - Individual, o por grupos.
 - Algunos forman parte de la nota de **evaluación continua**; otros no.
 - En cualquier caso, podéis pedir ayuda al profesor.
 - Son ejercicios cortos; se entregan en la misma sesión de clase.

[3] Clase de laboratorio

Martes de cada semana (12-13h)

- Un problema a realizar y entregar en *DOMjudge*
 - Dificultad similar a los problemas de examen.
 - Pensado para realizarse en una hora, pero no se exige entregarlo en ese momento.
 - Por parejas, fijadas de antemano.



Calendario de una semana

Las “semanas” de la asignatura no coinciden con las semanas naturales

	Jueves	Viernes		Martes	
<i>Semana n-1</i>	Material disponible en CV				
Semana n	Test de control Resolución problemas	Resolución problemas		Problema laboratorio	

Evaluación



Evaluación

30%

Evaluación
continua

70%

Examen final



Evaluación

30%

70%

Examen final

- Examen en laboratorio.
- Problemas que se entregan en *DOMjudge*.
- Se corregirá el examen independientemente del veredicto del juez.
 - No es necesario veredicto ACCEPT.
- **Para aprobar la asignatura se necesita una calificación ≥ 5 en el examen.**

Evaluación



Evaluación
continua

Hay dos modalidades de evaluación continua:

- **Modalidad A:** «*Quiero venir a clase*»
- **Modalidad B:** «*No quiero venir a clase*»



Evaluación continua – Modalidad A



Problemas de
laboratorio

- Problemas que se entregan en *DOMjudge*.
 - Uno por semana (con alguna excepción).
- Se requiere veredicto ACCEPT, pero puedes pedir ayuda al profesor si se te resiste.
- Se calcula la media de todas las calificaciones, descartando las dos notas más bajas.

Evaluación continua – Modalidad A



Actividades en aula

- Test de control y, ocasionalmente, ejercicios en grupos de cuatro personas que se realicen durante las clases de teoría.
- Se calcula la media de todas las calificaciones obtenidas, descartando previamente las tres notas más bajas.

Evaluación continua – Modalidad A



Actividades en aula

Fechas de test de control (orientativas)

- 1 de febrero
- 8 de febrero
- 15 de febrero
- 22 de febrero
- 29 de febrero
- 7 de marzo
- 14 de marzo
- 21 de marzo
- 4 de abril
- 11 de abril

Evaluación continua – Modalidad A



Actividades en aula

Fechas de ejercicios en grupos de cuatro personas (orientativas)

- 16 de febrero
- 23 de febrero
- 15 de marzo
- 5 de abril
- 19 de abril

Evaluación continua – Modalidad A



Asistencia y
participación

¿Qué se considera «participación»?

- Trabajar en las actividades que se realicen ese día en clase.
- Se aplica a las clases de teoría y a las de laboratorio.

Evaluación continua – Modalidad A



Asistencia y
participación

¿Qué NO se considera «participación»?

- Estar sentado sin hacer nada.
- Consultar redes sociales, mensajes, etc.
- Consultar cualquier página que no tenga relación con la asignatura.
- Hacer actividades que no tengan relación con la asignatura, aunque sean de otras asignaturas.

Evaluación continua – Modalidad A



Asistencia y
participación

¿Y si termino/terminamos la actividad antes que los demás compañeros?

- Puedes decírselo al profesor, para avanzar con la siguiente actividad que haya en esa clase, o bien,
- Si no hay más actividades durante ese día, puedes irte.

Evaluación continua – Modalidad A



Asistencia y
participación

¿Cómo se puntúa?

- Se controla la asistencia en el periodo 1 de febrero – 26 de abril.
- Para cada clase, 1 punto si hay participación, 0 si no la hay.
- Media de todas las notas, descartando las 3 más bajas.

Evaluación continua – Modalidad B



Evaluación
continua

Modalidad B: «*No quiero venir a clase*»

- Ejercicio en laboratorio en la clase del día 7 de mayo.
- Examen escrito en la clase del día 9 de mayo.
 - En las preguntas tipo test, los fallos restan puntuación.

Evaluación continua



Evaluación
continua

¿Cómo se elige modalidad?

- Si eliges *Modalidad A*: no tienes que hacer nada.
- Si eliges *Modalidad B*: tienes que comunicárselo por correo electrónico al profesor antes del **1 de febrero**.

A partir del 1 de febrero **no se permiten cambios de modalidad**, salvo en situaciones sobrevenidas y muy justificadas.

Requisitos



Requisitos

- **Fundamentos de programación 1 y 2**
 - Punteros
 - Manejo de memoria dinámica (new y delete)



Requisitos

- Fundamentos de programación 1 y 2
- **Tecnología de la programación 1**
- Encapsulación mediante clases y objetos

Herencia no lo vamos a utilizar mucho



Requisitos

- Fundamentos de programación 1 y 2
- Tecnología de la programación 1
- **Fundamentos de Algoritmos**
 - Costes asintóticos en tiempo y espacio
 - Recursión
 - Búsqueda binaria



Contenidos de la asignatura



Objetivos

- Distinguir entre el modelo conceptual de un tipo de datos y su representación.

Concepto clave: **Tipo Abstracto de Datos**



Objetivos

- Distinguir entre el modelo conceptual de un tipo de datos y su representación.
- Conocer los TADs más habituales en programación, y cómo se implementan.

- **Listas**

- **Pilas**

Tipos de darts lineales

- **Colas**

- **Conjuntos**

pos de datos asociativos

- **Diccionarios**

Objetivos

- Distinguir entre el modelo conceptual de un tipo de datos y su representación.
- Conocer los TADs más habituales en programación, y cómo se implementan.
- Identificar las estructuras de datos más adecuadas para resolver un determinado problema de modo eficiente.

Ello exige conocer el **coste** de las operaciones.

Objetivos

- Distinguir entre el modelo conceptual de un tipo de datos y su representación.
- Conocer los TADs más habituales en programación, y cómo se implementan.
- Identificar las estructuras de datos más adecuadas para resolver un determinado problema de modo eficiente.
- Poder extender un TAD añadiendo operaciones propias.

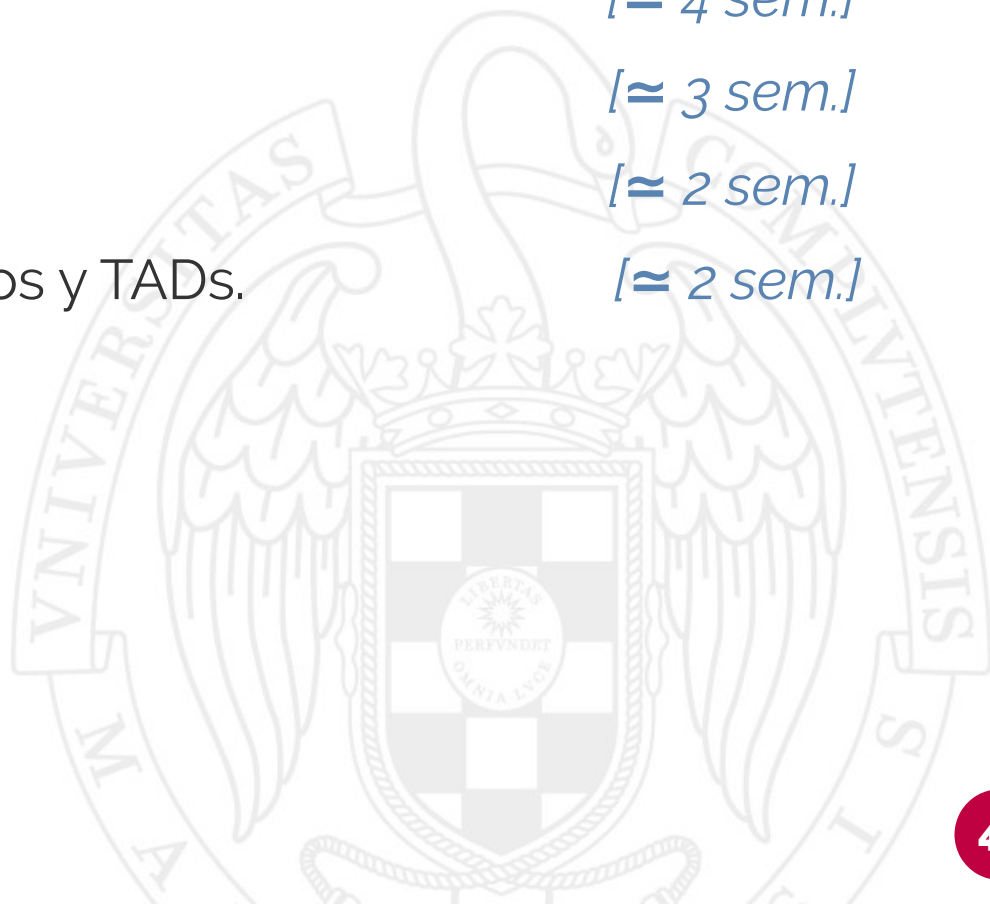
Objetivos

- Distinguir entre el modelo conceptual de un tipo de datos y su representación.
- Conocer los TADs más habituales en programación, y cómo se implementan.
- Identificar las estructuras de datos más adecuadas para resolver un determinado problema de modo eficiente.
- Poder extender un TAD añadiendo operaciones propias.
- Conocer los mecanismos de C++ para la implementación de TADs.

- **Clases y objetos**
- **Memoria dinámica**
- **Genericidad**
- **Sobrecarga**

Contenidos

- 1) Introducción a los tipos abstractos de datos. *[≈ 1 sem.]*
- 2) Tipos de datos lineales. *[≈ 4 sem.]*
- 3) Tipos de datos arborescentes. *[≈ 3 sem.]*
- 4) Diccionarios. *[≈ 2 sem.]*
- 5) Aplicaciones de estructuras de datos y TADs. *[≈ 2 sem.]*



Recursos



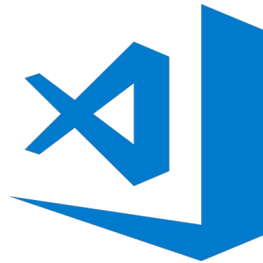
Software recomendado

Linux

Compilador g++



MS Visual Studio Code
Plugin C/C++



Windows

Visual Studio 2019



Podemos tenerlo bien y como el juez usa linux nos ponga wrong-answer

o bien

MinGW (g++)
MS Visual Studio Code
Plugin C/C++



Páginas web

- Juez para entrega de prácticas en laboratorio - <https://ed.fdi.ucm.es>
 - Cada estudiante tendrá una cuenta.
 - También pueden proponerse problemas de autoevaluación.
- Acepta El Reto - <https://www.aceptaelreto.com/>
 - Se propondrán problemas de autoevaluación.
- Compiler Explorer: <https://godbolt.org/>



Bibliografía recomendada

- Referencia de la librería estándar de C++

<https://en.cppreference.com/w/>

Estará disponible durante el examen final.

C++ reference		
C++98, C++03, C++11, C++14, C++17, C++20		
Compiler support Freestanding implementations	Concepts library (C++20)	Iterators library
Language	Diagnostics library	Ranges library (C++20)
Basic concepts	General utilities library	Algorithms library
C++ Keywords	Smart pointers and allocators	Numerics library
Preprocessor	Date and time	Common math functions
Expressions	Function objects — hash (C++11)	Mathematical special functions (C++17)
Declaration	String conversions (C++17)	Numeric algorithms
Initialization	Utility functions	Pseudo-random number generation
Functions	pair — tuple (C++11)	Floating-point environment (C++11)
Statements	optional (C++17) — any (C++17)	complex — valarray
Classes	variant (C++17) — format (C++20)	
Templates	Strings library	Input/output library
Exceptions	basic_string	Stream-based I/O
Headers	basic_string_view (C++17)	Synchronized output (C++20)
Named requirements	Null-terminated strings:	I/O manipulators
Feature test macros (C++20)	byte — multibyte — wide	Localizations library
Language support library	Containers library	Regular expressions library (C++11)
Type support — traits (C++11)	array (C++11) — vector	basic_regex — algorithms
Program utilities	map — unordered_map (C++11)	Atomic operations library (C++11)
Relational comparators (C++20)	priority_queue — span (C++20)	atomic — atomic_flag
numeric_limits — type_info	Other containers:	atomic_ref (C++20)
initializer_list (C++11)	sequence — associative	Thread support library (C++11)
	unordered associative — adaptors	Filesystem library (C++17)
Technical specifications		
Standard library extensions (library fundamentals TS)		
resource_adaptor — invocation_type		
Standard library extensions v2 (library fundamentals TS v2)		
propagate_const — ostream_joiner — randint		
observer_ptr — detection idiom		
Standard library extensions v3 (library fundamentals TS v3)		
scope_exit — scope_fail — scope_success — unique_resource		
Concurrency library extensions (concurrency TS)		
Concepts (concepts TS)		
Ranges (ranges TS)		
Transactional Memory (TM TS)		
External Links — Non-ANSI/ISO Libraries — Index — std Symbol Index		

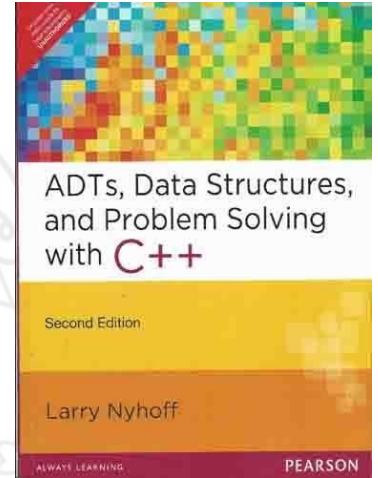
Bibliografía recomendada

Larry Nyhoff

*ADTs, Data Structures and Problem Solving with C++
(2nd edition)*

Pearson (2011)

<https://ucm.on.worldcat.org/oclc/1026080575>



- Cubre la mayoría de los temas.
- Incluye introducción a la programación orientada a objetos con C++.

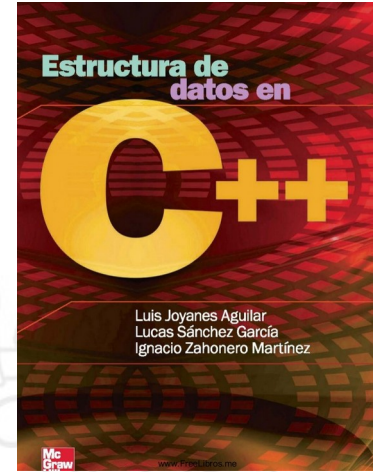
Bibliografía recomendada

Luis Joyanes, Lucas Sánchez, Ignacio Zahonero

Estructura de datos en C++

McGraw Hill (2007)

<https://ucm.on.worldcat.org/oclc/427540716>



- Cubre bien los temas relativos a Estructuras de Datos y ADTs
- En lo que concierne a C++, un poco más desfasado.
- Disponible *online* a través de la web de la biblioteca de la UCM.

Bibliografía recomendada

N. Martí, Y. Ortega, A. Verdejo

Estructuras de Datos y Métodos Algoritmos

Garceta, D. L. (2013)

<https://ucm.on.worldcat.org/oclc/1025987082>



- Los ejercicios están resueltos.
- Algunos son sobre especificación algebraica (no entra en el temario).
- Los algoritmos se expresan en pseudocódigo (no en C++).

Atención al estudiante

- ✓ Atención de dudas por correo: montenegro@fdi.ucm.es

Si la duda es sobre un envío que has hecho al juez, indica tu **nombre de usuario** y el **ID** del envío.

- ✓ **Tutorías:**

- Martes 14-15h, 16-17h,
- Jueves 10-12h,
- Cualquier otro horario, previa petición de hora.

- ✗ **NO** preguntar dudas a través de:

- Mensajes privados en el CV,
- Sistema de *clarifications* de DOMjudge.

¿Qué haremos antes del 1 de febrero?

- **25 de enero**
 - Ejercicio práctico en el aula
 - No puntúa para la evaluación continua
- **30 de febrero**
 - Clase en aula (no en laboratorio)
 - Test de iniciación
 - Conceptos básicos sobre costes y memoria dinámica.
 - No puntúa para la evaluación continua.

