

	<b>PROGRAMACIÓN II</b>		<b>3.4.074</b>
<b>Departamento al que pertenece</b>		<b>Director</b>	
<b>Tecnología Informática</b>		<b>Ing. Anibal Freijo</b>	
<b>Carga horaria</b>		<b>Fecha de aprobación en el Consejo de Facultad y N° de Acta</b>	
68 horas		03/08/2021 CF FAIN N° 443	
<b>Carrera(s) en la que se dicta</b>		<b>Código(s) Carrera(s)</b>	
Tecnicatura Universitaria en Desarrollo de Software Licenciatura en Gestión de Tecnología de la Información Licenciatura en Bioinformática Ingeniería en Informática Ingeniería en Informática		1111 13116 5910 1608 1621	
<b>Código(s) Correlativa(s) Precedente(s)</b>	<b>Código(s) Correlativa(s) Subsiguiente(s)</b>	<b>Código(s) Carrera(s)</b>	
3.4.071	3.4.077	13116 / 1111 / 1608 / 1621 13116 / 1111 / 1608 / 1621	
<b>Firmas</b>			
<b>Aprobación del Director de Departamento emisor.</b>		<b>Aprobación Decano(s)</b>	
<div style="text-align: center;">   <b>Ing. Anibal Freijo</b>  <b>Director del Departamento de Informática</b>  <b>Facultad de Ingeniería y Cs. Exactas</b>  <b>Fundación UADE</b> </div>		<div style="text-align: center;">   <b>Dr. Federico Prada</b>  <b>Decano a cargo</b>  <b>Facultad de Ingeniería y Cs. Exactas</b>  <b>Fundación UADE</b> </div>	

## I. Fundamentos de la materia

En las materias previas se adquirieron los conocimientos básicos de programación y de diseño de programas. Esta asignatura incorpora los conceptos de Tipos de Datos Abstractos (TDA), sus especificaciones, sus implementaciones y aplicaciones. También incluye los conocimientos para evaluar la complejidad espacial y temporal de las distintas implementaciones.

Se complementa con la materia siguiente, Programación III, sobre estructuras de datos y técnicas de diseño de algoritmos.

Estos conceptos serán fundamentales a la hora de enfrentarse con el aprendizaje de los nuevos paradigmas de programación en las siguientes materias, donde continuará la evolución desde las aplicaciones orientadas a objetos a las orientadas a componentes, desarrollando aplicaciones con tecnologías actuales.

## II. Objetivos

Que el estudiante sea capaz de:

- Comprender y saber utilizar Tipos de Datos Abstractos (TDAs).
- Comprender la eficiencia de una implementación de un TDA.
- Expresar TDAs en términos de un lenguaje de programación.
- Entender, evaluar y seleccionar las mejores opciones de Tipos de Datos Abstractos para solucionar diferentes tipos de problemas.
- Resolver problemas de programación expresados en términos de tipos de datos abstractos.
- Encarar el desarrollo de una aplicación pequeña, empleando las estructuras de datos adecuadas.

## III. Contenidos

### **Contenidos mínimos**

Tipos de Datos Abstractos (TDAs). Especificación de TDAs. Estructuras de datos básicas. Descripción de problemas utilizando tipos abstractos de datos. Análisis de eficiencia de algoritmos. Complejidad temporal y espacial. Utilización de un lenguaje de programación para las implementaciones de TDAs. Resolución de problemas.

### **Contenidos conceptuales**

#### Unidad I: Introducción

Introducción a los Tipos de Datos Abstractos (TDA). ¿Qué es un TDA? Características. Datos y operaciones relacionadas. Ejemplos de TDAs: Pila, Cola, Cola con Prioridad, Conjunto, Diccionario. Especificación de un TDA. Notación utilizada.

#### Unidad II: Nociones de Implementaciones de TDAs

Nociones de implementación de un TDA. De la especificación a la implementación. Implementación de los TDAs vistos utilizando estructuras de datos estáticas. Ventajas y limitaciones. Complejidad temporal y espacial de las operaciones. Ordenes de complejidad. Criterios para la elección de una implementación en función del costo de las operaciones asociadas.

#### Unidad III: Familiarización con un Lenguaje de Programación

Utilización de un lenguaje de programación concreto para expresar la especificación e implementación de un TDA. Ventajas y limitaciones.

#### Unidad IV: Implementación de TDAs mediante estructuras de datos dinámicas

Implementación de los TDAs vistos utilizando estructuras de datos dinámicas. Comparación entre los costos de las implementaciones estáticas y las dinámicas. Resolución de problemas combinando distintos TDAs. Resolución de problemas combinando distintas implementaciones de un mismo TDA.

Unidad V: El TDA Árbol

TDA Árbol Binario (AB) y el Árbol Binario de Búsqueda (ABB). Especificación del TDA ABB. Implementaciones del TDA ABB. Ejemplos de utilización de un ABB. Resolución de problemas.

Unidad VI: El TDA Árbol Genérico, AVL y B.

TDA Árbol Genérico (n-ario), Árbol Balanceado (AVL) y Árbol B (B). Especificación del TDA n-ario, AVL y B. Implementaciones del TDA n-ario, TDA AVL y B. Ejemplos de utilización de árboles n-ario, AVL y B. Resolución de problemas.

Unidad VII: El TDA Grafo

TDA Grafo. Definición: concepto de grafo, grafos dirigidos, grafos no dirigidos, caminos, ciclos, conectividad, etc. - Especificación del TDA Grafo. Implementaciones estáticas y dinámicas del TDA Grafo. Resolución de problemas.

**Contenidos procedimentales**

- Expresión de soluciones en términos de TDAs
- Definición de nuevos TDAs
- Determinación de la mejor implementación de un TDA para un problema dado
- Implementación de TDAs con un lenguaje de programación.
- Medición de eficiencia temporal y espacial de un programa.

**Contenidos actitudinales**

- Valoración de la calidad, legibilidad y eficiencia de un programa
- Proactividad para resolver problemas
- Identificación de niveles de abstracción de una solución.

**IV. Estrategias de enseñanza**

Exposiciones dialogadas para la enseñanza de los conceptos, combinando exposición docente con participación activa de los alumnos.

Presentación de ejemplos para complementar la exposición dialogada. Resolución de problemas por parte de los alumnos a efectos de motivar la aplicación de conocimientos.

**V. Recursos**

- Aula tradicional con medios para la proyección de contenidos
- Software necesario para la realización de los trabajos prácticos

**VI. Modalidad de Evaluación**

La asignatura PROGRAMACIÓN II se aprueba con Final Obligatorio.

Para aprobar la asignatura el alumno deberá cumplir con la aprobación de dos instancias de exámenes: la cursada y el examen final, las cuales podrán ser individuales o grupales y escritos y/u orales. Requisitos académicos obligatorios para aprobar la cursada: a) Cumplir con un mínimo del 75% de asistencia a clase. b) Aprobar cada una de las evaluaciones parciales y/o recuperatorios previstos con una calificación mínima de 4 (cuatro) puntos. c) Aprobar los trabajos prácticos obligatorios (TPO) si los hubiere. Quienes cumplan con todos los requisitos indicados, quedarán habilitados para rendir el examen final de la asignatura en los 11 (once) turnos de exámenes finales consecutivos posteriores a la aprobación de la cursada. Se consignará como nota final el promedio simple entre la nota de aprobación de la cursada (promedio de las calificaciones en las evaluaciones parciales/TPs aprobados) y la calificación obtenida en el examen final regular. En el caso que el alumno haya aprobado las instancias de evaluación y no requiera recuperar, podrá optar por rendir el examen final regular en la fecha prevista para el examen

recuperatorio o bien en la fecha prevista para el examen final regular (una de las dos). Los alumnos que rindan el examen final en la etapa de previos, la nota final a consignarse será exclusivamente la obtenida en dicha instancia de evaluación. Los actos de deshonestidad académica o cualquier situación de indisciplina serán sancionados según el régimen disciplinario correspondiente.

**NORMAS DE SEGURIDAD:** El trabajo en laboratorios y talleres debe llevarse a cabo respetando las normas de seguridad obligatorias. La aprobación de la cursada/materia estará sujeta al cumplimiento de las mismas, ya que son el principal factor de riesgo en las actividades de los alumnos, docentes, investigadores o técnicos.  
Utilizar siempre los Equipos de Protección Individual que se requiera (consultar procedimientos o protocolos de trabajo), por ejemplo protección ocular (anteojos gafas/pantallas faciales), guantes de vinilo y guardapolvo.

## VII. Bibliografía

### Básica

AHO, J.; HOPCROFT, John E. y ULLMAN, Jeffrey D.. Estructuras de Datos y Algoritmos. 1988. Wilmington : Addison Wesley Iberoamericana. 438 p. ISBN: 02016402449684443455.

WEISS, Mark Allen. Estructuras de datos en JAVA. 2004. Madrid: Addison Wesley. 740 p. ISBN: 9788478290352

### Complementaria

AHO, A. ULLMAN, J. Foundation of Computer Science. C Edition. Computer Science Press. 1995. ISBN-13: 978-0716782841 ISBN-10: 0716782847

ECKEL, Bruce. Thinking in Java. 3rd edition. [en línea] [consulta: 24 de feb. 2014]  
<http://www.mindviewinc.com/Books/downloads.html>

GOODRICH, M.; TAMASSIA, R. Estructura de Datos y Algoritmos en Java. 2ed. 2001. New York: John Wiley. & Sons Inc. 738 p. ISBN: 0-471-73884-0

HOROWITZ, Ellis ; SAHNI, Sartaj y RAJASEKARAN, Sanguthevar. Computer Algorithms / C++. 1ra Edición, 1996. Computer Science Press. ISBN978-0716783152

LEWIS J. y CHASE, J.. Estructuras de Datos con Java. Diseño de Estructuras y Algoritmos. PEARSON EDUCACION, 2006. ISBN 9788420550343

## VIII. Cronograma

Clase	Actividad/contenido
1	Introducción Tipos de Datos Abstractos (TDA) . Especificaciones de TDA - Notación utilizada - Fundamentos de Java.
2	Implementación estática de un TDA.
3	TDA Cola con Prioridad. Implementación. TDA Conjunto
4	TDA Diccionario. Implementaciones dinámicas.
5	Resolución de problemas combinando distintos TDA
6	Comparación de implementaciones estáticas y dinámicas.
7	Criterios para seleccionar implementaciones. Eficiencia Complejidad espacial y temporal.
8	Conceptos Generales sobre Árboles. TDA Árbol Binario de Búsqueda (ABB) - Especificación, Implementación, Recursividad, Recorridos.
9	Ejemplos de utilización - Resolución de problemas.
10	TDA Árbol B - Especificación y comportamiento.
11	Ejemplos de utilización - Resolución de problemas.
12	TDA Grafo: Definición (concepto de grafo, grafos dirigidos, grafos no dirigidos) - Especificación e Implementación.
13	Ejercitación de Grafos.

14	Resolución de problemas.
15	Recuperatorio
16	Entrega Final TPO
17	Final Regular