

000027

Estructuras de datos

INSTITUTOS/S: **INSTITUTO DE TECNOLOGÍA E INGENIERÍA**

CARRERA/S: **Licenciatura en Informática**

RESPONSABLE DE LA ASIGNATURA Y EQUIPO DOCENTE:
Sergio Alberto González

Martín Pustilnik, Oscar Filevich

AÑO: 2025

CRÉDITOS: 7

CARGA HORARIA DE INTERACCIÓN PEDAGÓGICA: 96

CARGA HORARIA TOTAL: 175

CÓDIGO DE LA MATERIA EN SIU: 752

000027

1. Fundamentación

El perfil del egresado de la carrera apunta a un profesional de la construcción de software con parámetros de calidad, tanto en etapas de diseño como de programación, con una base conceptual sólida que permita adaptarse a proyectos de distinta índole y favorecer el trabajo en equipo y colaborativo. En este contexto, la materia toma un importante rol, ya que, en esta se adquieren conceptos centrales que luego se utilizarán a lo largo de la carrera y en la práctica profesional.

2. Propósitos y/u objetivos

- *Comprender los conceptos de programación estructurada y modular. Comprender el concepto de dato, tipos de datos primitivos y estructuras de datos, y su importancia e interrelación con la resolución algorítmica de un problema.*
- *Conocer la interfaz de una estructura de datos y su utilización para la resolución de problemas. Incorporar los conceptos de ámbito y de pasaje de parámetros por valor o por referencia.*
- *Resolver problemas mediante programas recursivos, y entender la diferencia entre una solución recursiva y una iterativa. Comprender el concepto de asignación dinámica de memoria y escribir programas que realicen un manejo dinámico explícito de memoria en forma adecuada.*
- *Conocer las estructuras de datos básicas y su interfaz: Vectores, matrices, pilas, colas, listas, árboles, diccionarios, hashing, grafos, etc. Conocer el funcionamiento interno de estructuras de datos básicas principios básicos del diseño de la interfaz.*
- *Entender la noción de implementación de una estructura de datos, y de su eficiencia, y ser capaz de implementar las interfaces vistas anteriormente con alternativas variadas en eficiencia. Familiarizarse con las tareas de compilar y vincular programas para lograr un ejecutable.*

000027

- *Incorporar la elección de la estructura de datos como una parte de la solución de un problema, en relación con la resolución algorítmica y la programación en computadora.*
- *Utilizar e internacionalizar buenas prácticas de programación, en relación con la práctica profesional, trabajo en equipo y generación de código legible y portable.*

3. Contenidos mínimos

Programación estructurada y modular. Programas recursivos. Representación de datos en memoria. Paso de variables por valor y por referencia. Tipos abstractos de datos. Estructuras de datos. Estructuras contenedoras: Vectores, matrices, pilas, colas, lista, diccionarios, árboles y grafos. Implementación de estructuras de datos estáticas. Uso dinámico de memoria. Listas y árboles implementados con punteros. Análisis, eficiencia e implementación de estructuras de datos. Algoritmos de recorrido, búsqueda y ordenamiento. Nociones básicas de algoritmos sobre grafos.

4. Carga horaria

<i>Créditos</i>	<i>Interacción pedagógica</i>	<i>Trabajo Autónomo</i>	<i>Total</i>
<i>7</i>	<i>Teoría: 32 Práctica: 64</i>	<i>79</i>	<i>175</i>

4.1 Trabajo autónomo del/la estudiante

<i>Actividad</i>	<i>Carga horaria</i>
<i>Lectura de contenidos teóricos apuntes o bibliografía.</i>	<i>20</i>
<i>Desarrollo de los ejercicios prácticos propuestos.</i>	<i>24</i>
<i>Revisar los videos teóricos.</i>	<i>10</i>
<i>Preparar temas para parcial.</i>	<i>25</i>

5. Programa analítico

UNIDAD 1: *Introducción.*

000027

Incorporar conceptos básicos necesarios para escribir, compilar y ejecutar un programa. Utilizar reglas sintácticas. Tipos de datos primitivos y referenciales. Definición de variables. Operadores. Estructuras de control. Funciones y paso de parámetros. Ámbito de variables. Sobrecarga de operadores. Proporcionar ejemplos y realizar ejercicios.

UNIDAD 2: *Tipos de datos abstractos.*

Incorporar el concepto de abstracción de datos y encapsulamiento de datos. Incorporar el concepto de tipo abstracto de dato (TDA). Diferenciar entre tipo de dato y tipo abstracto de dato. Realizar requerimientos y diseño de un TDA. Trabajar con TDAs.

UNIDAD 3: *Funciones recursivas, arreglos unidimensionales y bidimensionales.*

Incorporar principio de recursión. Funcionamiento de recursión. Trabajar problemas y ejercicios con soluciones recursivas. Trabajar con arreglos unidimensionales y bidimensionales, representación en memoria, creación, recorrido, inserción y eliminación de elementos.

UNIDAD 4: *Pilas dinámicas, definición y terminología.*

Incorporar definición y terminología de pilas. Trabajar con el TDA pila. Operaciones básicas. Realizar implementación dinámica mediante listas.

UNIDAD 5: *Colas dinámicas, definición y terminología.*

Incorporar definición y terminología de colas. Trabajar con el TDA cola. Operaciones básicas. Realizar implementación dinámica mediante listas. Colas simples y circulares.

UNIDAD 6: *Diccionarios.*

Incorporar definición y terminología de diccionarios. Trabajar con TDA Diccionario. Operaciones básicas. Implementación utilizando conjuntos y tuplas.

UNIDAD 7: *Listas.*

000027


Incorporar definición y terminología de listas. Trabajar con TDA lista. Operaciones básicas. Implementación dinámica utilizando punteros. Listas simples enlazadas, dobles enlazadas y circulares.

UNIDAD 8: Árboles.

Incorporar definición y terminología de árbol. Árboles binarios. Árboles binarios de búsqueda. Realizar implementación utilizando punteros. Recorridos e implementaciones recursivas. Inserción, eliminación, modificación y búsqueda de elementos. Árboles balanceados. Árboles AVL. Árboles de M-vías. Trabajar con árboles y sus aplicaciones.

UNIDAD 9: Grafos.

Incorporar definición y terminología de grafos. Realizar recorridos básicos, aplicaciones y ejemplos.

6. Ejes y enunciados Multidimensionales y transversales

<i>Eje</i>	<i>Nivel de logro de Aprendizaje</i>	<i>Acciones de enseñanza</i>
<i>C1. Identificación, formulación y resolución de problemas de informática</i>	<i>Bajo</i>	<i>Aprendizaje basado en problemas mediante las clases prácticas, resolución de ejercicios y problemas.</i>
<i>C2. Concepción, diseño y desarrollo de proyectos de informática</i>	<i>No aporta</i>	-
<i>C3. Gestión, planificación, ejecución y control de proyectos de informática</i>	<i>No aporta</i>	-
<i>C4. Utilización de técnicas y herramientas de aplicación en la informática</i>	<i>No aporta</i>	-
<i>C5. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas</i>	<i>No aporta</i>	-
<i>C6. Fundamentos para el desempeño en equipos de trabajo</i>	<i>No aporta</i>	-

000027

C7. Fundamentos para la comunicación efectiva	No aporta	-
C8. Fundamentos para la acción ética y responsable	No aporta	-
C9. Fundamentos para evaluar y actuar en relación con el impacto social de su actividad en el contexto global y local	No aporta	-
C10. Fundamentos para el aprendizaje continuo	Bajo	Clases expositivas dialogadas, favoreciendo la participación de las/los estudiantes y haciendo foco en que los temas que se abordan en la materia se encuentran en constante desarrollo.
C11. Fundamentos para la acción emprendedora	No aporta	-

7. Bibliografía y recursos

7.1 Bibliografía obligatoria

Mc. Silvia Guardati Buemo (2023). “Colecciones de datos + algoritmos en Python. De cero al infinito”- 1ra Ed.- Editorial Alfaomega.

Joyanes Aguilar, Luis. (2020). "Fundamentos de programación. algoritmos y estructuras de datos". 5ta Edición, Mc Graw-Hill.
 Disponible en : <https://books.ms/main/B2E6E8928FBB13638A8CFAC93F0030B3>

Robert Lafore, Alan Broder And John Canning (2022). “Data Structures and Algorithms in Python”- 1ra Ed- Editorial Pearson.
 Disponible en:
<https://dokumen.pub/data-structures-amp-algorithms-in-python-1nbsped-013485568x-9780134855684-9780137916191-9780134855912.html>

Brow, Martin C (2019). “Python: The Complete Reference” -1ra ed- Editorial McGraw Hill.

7.2 Bibliografía optativa

Mark Allen Weiss. (1997) “Data Structures and Algorithms in C”. Addison – Wesley (2da Edición).

000027

Chris Okasaki. (2003) "Purely Functional Data Structures", Cambridge University Press.

Alfred Aho & Jeffrey Ullman. (1983) "Data Structures and Algorithms" -1ra Ed-. Editorial Addison – Wesley

7.3 Recursos

Los recursos incorporados se encuentran tanto en el campus como en plataformas asincrónicas de discusión y consulta.

En el campus: El material soporte de cada clase. Consiste en las presentaciones de las teóricas y la bibliografía adicional correspondiente.

Material audiovisual. Se pondrán a disposición de las/los estudiantes videos explicativos de las temáticas desarrolladas en las clases.

Foros proactivos de discusión. Las/Los docentes propondrán foros de discusión por Unidad Temática.

Ejercicios y actividades de autoevaluación. Se ponen a disposición de las/los estudiantes ejercicios prácticos y cuestionarios en formato múltiple choice correspondientes a las diferentes clases teóricas.

En plataformas asincrónicas (por ejemplo, Discord): Se plantea un foro de discusión y consulta constante, donde las/los estudiantes puedan volcar dudas sobre las actividades prácticas a lo largo de la semana, previo a la clase práctica de resolución de ejercicios.

8. Metodología de enseñanza

8.1 Modalidades u opciones pedagógicas

Las clases se desarrollan en una modalidad mixta presencial - virtual en ciclos semanales. Utilizando el soporte de la plataforma virtual del Campus de la Universidad se articulan clases teóricas virtuales asincrónicas cortas, en las cuales se vierten los conceptos básicos de cada uno de los temas en cuestión, con clases teórico - prácticas de repaso, virtuales sincrónicas con ejemplos prácticos y en la que se favorece la participación de los/las estudiantes. Ambas clases virtuales quedan disponibles en el Campus virtual como material de repaso y consulta de forma permanente. Se finalizan cada uno de los temas con una clase práctica

000027

presencial en la que se resuelven situaciones problemáticas en el pizarrón y se plantean casos de uso en situaciones reales.

La mayor cantidad de tiempo se dedica a la parte práctica, con la resolución de problemas en el pizarrón de forma constructiva con participación de los/las estudiantes, permitiendo la discusión e interacción y principalmente con la realización de ejercicios por parte de los/las estudiantes en el laboratorio de computación.

Cada unidad perteneciente a los contenidos mínimos posee una guía de ejercicios asociada y es de carácter obligatorio la realización de actividades de autoevaluación semanales a través del campus virtual. Para la implementación de los ejercicios se utilizan lenguajes de programación de acuerdo con el estado del arte de la tecnología, que se adecuen a los propósitos pedagógicos de la materia.

Las guías de ejercicios/autoevaluación están redactadas considerando un lenguaje inclusivo que no prioriza un género sobre otro.

Plan de trabajo en el campus.

Se utiliza la plataforma virtual que provee la Universidad para el armado de clases teóricas virtuales, para la exposición de algunos temas, con el fin de facilitar el aprendizaje y uso de herramientas informáticas, potenciando además la autonomía, la responsabilidad y la comunicación interactiva en los estudiantes.

Los estudiantes contarán para la asignatura con una plataforma Moodle, a través del campus virtual de la UNAHUR. En dicha plataforma, los estudiantes tienen disponible desde el inicio de clases: el programa, el cronograma y la bibliografía de la materia. Previo al dictado presencial de las Unidades, se habilita para cada una de éstas:

- El material soporte de cada clase. Consiste en las presentaciones de las teóricas y la bibliografía adicional correspondiente.*
- Material audiovisual. Se pondrán a disposición de las/los estudiantes videos explicativos de las temáticas correspondientes, para mejorar el aprendizaje de los contenidos teóricos desarrollados en las clases. En caso de modalidad enteramente virtual se pondrán a disposición videos correspondientes a todas las clases teóricas de la materia.*
- Foros proactivos de discusión. Las/Los docentes propondrán foros de discusión por Unidad Temática, mediante preguntas disparadoras, artículos publicados, noticias u otras herramientas que fomenten la participación activa de las/los estudiantes. Los foros podrán ser utilizados también por las/los estudiantes para*

000027

la realización de consultas e intercambios con las/los docentes y/o entre ellas/ellos mismas/mismos.

- Ejercicios y actividades de autoevaluación. Se ponen a disposición de las/los estudiantes ejercicios prácticos y cuestionarios en formato multiple choice correspondientes a las diferentes clases teóricas. Esto permitirá a las/los estudiantes ejercitarse y evaluar el grado de aprendizaje adquirido. Se podrá realizar en la cantidad de intentos que cada estudiante desee, dado que el resultado obtenido no será tenido en cuenta para su consideración por parte del cuerpo docente de la asignatura.

8.2 Formación Práctica

Durante la resolución de la totalidad de las actividades propuestas, se exige a las/los estudiantes mantener y respetar buenas prácticas de programación, en relación con la práctica profesional, trabajo en equipo y generación de código legible y portable. La idea es que, con la utilización de estas prácticas, cada estudiante las internalice en su uso diario y que pasen a formar parte de su práctica profesional.

A modo de complementar e integrar parte de los temas que forman parte del programa de la materia, se propone la realización de un trabajo práctico grupal (carga horaria total: 6 hs).

Trabajo Práctico 1. Algoritmos de Búsqueda y Ordenamiento: En este trabajo práctico se propone la integración de los temas de la primera parte del programa, abordando los problemas de búsqueda y ordenamiento y utilizando el tipo de dato arreglo como modelo.

Sus objetivos son: Integrar los conceptos de recursividad y arreglos. Comprender los conceptos de algoritmos de búsqueda y ordenamiento. Implementar los algoritmos. Comparar los algoritmos según su complejidad temporal, para el caso promedio.

9. Evaluación y régimen de aprobación

9.1 Modalidad de evaluación

La calificación de las/los estudiantes se compone del promedio de las notas de los 2 parciales (cada uno con sus respectivos recuperatorios), haciendo foco en las

000027

resoluciones conceptuales y otorgando menor peso a errores de codificación, intentando no evaluar los conocimientos de un lenguaje de programación en particular. Se toma en cuenta como nota de concepto el trabajo en clase y las entregas obligatorias de ejercicios de autoevaluación semanal.

9.2 Aprobación de la cursada

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos. Todas las instancias evaluativas deberán tener una instancia de recuperatorio. Podrán acceder a la administración de esta modalidad solo aquellos y aquellas estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial.

Siempre que se realice una evaluación de carácter recuperatorio, la calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

9.3 Acreditación de la materia

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

Promoción directa: *tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.*

Evaluación integradora: *tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos estudiantes que hayan aprobado lo cursado con una nota de entre cuatro (4) y seis (6) puntos.*

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrada, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia.

000027

La nota obtenida se promediará con la nota de la cursada.

Examen final: Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.