

Carrera: INGENIERIA EN INFORMATICA		
Asignatura: 3640 - Algoritmos y Estructuras de Datos		
Trayecto Programación		
Año académico 2023		
Responsable / Jefe de cátedra Lic. Renata Silvia Guatelli		
Carga horaria semanal 4 hs.	Carga horaria total 64 hs.	Créditos -----
Modalidad: Presencial		
Correlativas anteriores TOPICOS DE PROGRAMACION	Correlativas posteriores PARADIGMAS DE PROGRAMACION - PROGRAMACION CONCURRENTE	
Conocimientos necesarios -----		

Descripción de la asignatura

Esta asignatura introduce al estudiante en la implementación y desarrollo de Tipos de Datos Abstractos (TDAs). Presenta las estructuras de datos lineales (pilas, colas, listas) y arborescentes (árboles binarios); se profundiza en el diseño iterativo y en el diseño recursivo, tanto en el razonamiento sobre la corrección de un diseño dado como en la detección y mejora de soluciones ineficientes. Se presentan diversas implementaciones de estructuras de datos lineales y arborescentes. Se realizan implementaciones realistas de las mismas, usando primitivas independientes del tipo de dato, tanto estos sean tipos de datos simples como recursivos, para versiones estáticas como dinámicas. Se trabaja de forma tal que el estudiante comprenda su representación en memoria.

Metodología de enseñanza

Se utilizan metodologías activas de enseñanza, especialmente aprendizaje basado en problemas (ABP). Se presenta cada unidad temática introduciendo los conceptos fundamentales realizando analogías con ejemplos reales, que permite relacionar los contenidos de la materia con las herramientas habituales de trabajo.

Los contenidos de la asignatura se presentan de forma iterativa e incremental, de forma que le permitan al estudiante, construir sus propios procedimientos para resolver una situación problemática, lo que implica que sus ideas puedan verse modificadas y de esta forma siga construyendo nuevos conocimientos.

La materia tiene una fuerte carga práctica. En la misma se resuelven problemas, desarrollando pequeños sistemas, aplicando lo visto en las clases, trabajando de forma colaborativa, simulando un entorno de trabajo real.

Se motiva a los estudiantes en el uso de los foros de la plataforma MIEI y recursos de Teams, para la resolución de dudas, tanto de conceptos teóricos como prácticos, permitiendo desarrollar las capacidades de comunicación y afianzar el uso del lenguaje técnico. Además,

la cátedra cuenta con soporte digital de los contenidos, que los estudiantes pueden consultar luego de haber asistido a la clase.

Objetivos de aprendizaje

A través de esta asignatura, el estudiante habrá adquirido los conocimientos necesarios y suficientes para estar en condiciones de:

Objetivos Generales:

La materia se desarrolla teniendo en cuenta los siguientes objetivos generales:

- analizar, plantear y resolver situaciones problemáticas.
- organizar y planificar su trabajo.
- hacer transferencia de los conocimientos teóricos a la práctica.
- adquirir la capacidad de trabajar en equipo.
- identificar un problema a partir de una situación problemática presentada.
- diseñar un algoritmo eficiente para la resolución de una situación problemática analizada.
- desarrollar un algoritmo utilizando un lenguaje de programación.
- plantear casos de prueba de forma tal de ver los casos generales y particulares de cada situación problemática planteada.
- expresar los contenidos teóricos de la materia y su vinculación con situaciones de la vida real.
- aplicar los principios de la Programación Estructurada.
- detectar la fuerte vinculación de esta asignatura con materias de años anteriores y posteriores del plan de carrera.
- integrar grupos de trabajo, potenciando su propio aprendizaje a través de la interacción y cooperación con sus pares.
- utilizar con fluidez el lenguaje técnico relacionado con la materia.

Objetivos Específicos:

- Diseñar e implementar como TDAs, estructuras de datos lineales y árboles binarios, utilizando diferentes implementaciones independientes del dato que se almacene en dicha estructura.
- Desarrollar e implementar la solución que mejor se adapte de acuerdo con los requisitos establecidos.
- Seleccionar las estructuras de datos más adecuadas para la resolución de un problema dado.
- Detectar las ventajas y desventajas de la programación recursiva.

Contenidos mínimos

Estructuras de datos genéricas. Tipo de Dato Abstracto. Recursivos. Representación en memoria. Recursividad. Estructuras. Pila, Cola, Lista y Árbol. Estrategias de implementación. Diseño e implementación de variantes de usos y aplicaciones.

Competencias a desarrollar

Genéricas:

- Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.
- Utilización de técnicas y herramientas de aplicación en la ingeniería en sistemas de información / informática.
- Desempeño en equipos de trabajo.
- Comunicación efectiva.
- Actuación profesional ética y responsable.
- Aprendizaje continuo.
- Desarrollo de una actitud profesional emprendedora.
- Concepción, diseño y desarrollo de proyectos de ingeniería en sistemas de información / informática.

Específicas:

- Especificación, proyecto y desarrollo de software.
- Especificación, proyecto y desarrollo de sistemas de información.
- Establecimiento de métricas y normas de calidad de software.

Programa analítico	
Unidad 1	Tipo de dato abstracto (TDA) - Tipo recursivos de datos Concepto de Tipo abstracto de datos. Introducción al uso de tipo recursivos de datos. Gestión de memoria dinámica. Implementación de estructuras de datos enlazadas mediante tipos recursivos de datos. Algoritmos iterativos y recursivos para resolver problemas de búsqueda y recorrido en estructuras de datos enlazadas accediendo directamente a la representación basada en nodos y punteros a nodos.
Unidad 2	Estructura de datos Pila Asignación dinámica de memoria vs. asignación estática de memoria. Primitivas para el manejo de Pilas, compatibilidad de primitivas entre la implementación estática y dinámica de Pilas. Su relación con la recursividad.
Unidad 3	Estructura de datos Cola como un TDA Asignación dinámica de memoria vs. asignación estática de memoria. Primitivas para el manejo de Colas, compatibilidad entre la implementación estática y dinámica de Colas. Su relación con el "buffer" de teclado.
Unidad 4	Estructura de datos Lista como un TDA Primitivas para el manejo de Listas. Creación, inserción, ordenamiento, búsqueda, eliminación, etc., con asignación dinámica de memoria. Listas circulares, su importancia en la implementación de colas y su parecido y diferencia con la implementación de pilas. Listas doblemente enlazadas. Diferentes tipos de implementación. Funciones map, filter y reduce.
Unidad 5	Recursividad Concepto de recursión. Estructura de funciones recursivas. Ejecución de funciones recursivas. Envoltura para funciones recursivas. Tipos de recursividad. Eficiencia de la recursividad. Recursión versus Iteración.

Unidad 6	<p>Estructura de datos Árbol</p> <p>Recursividad. Árbol binario, creación, recorridas EnOrden, PreOrden y PosOrden. Árbol binario de búsqueda, su relación con la búsqueda binaria en arrays. Determinación de altura, y otras funciones. Árboles AVL y balanceados, determinación. Diferentes tipos de implementación.</p> <p>Concepto de archivos de índice (.idx) Implementación de índices sobre un árbol binario de búsqueda. Ventajas de implementar índices sobre árboles binarios de búsqueda.</p>
-------------	---

Planificación de actividades [15/16 semanas dependiendo del calendario académico universitario – UNLaM]					
Semana	Clase	Actividad	Tipo	Duración estimada	Unidad /des
Semana 1	1	Presentación de los docentes del curso, breve explicación de las pautas generales de la materia. Breve repaso de conceptos adquiridos en “Tópicos de Programación”, TDA, Memoria dinámica, funciones genéricas. Tipo recursivo de datos. Concepto de nodo.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	0 - 1
Semana 2	2	Estructura de datos Pila. Implementación como TDA sobre una estructura dinámica. Aplicaciones de Pilas, ejemplos sobre la práctica. Presentación del trabajo práctico.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	2
Semana 3	3	Estructura de datos Pila. Implementación como TDA sobre una estructura estática. Primitivas coherentes entre una implementación estática y dinámica. Aplicaciones de Pilas, ejemplos sobre la práctica.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	2
Semana 4	4	Aula: Estructura de datos cola. Implementación como TDA sobre una estructura dinámica. Aplicaciones de Colas, ejemplos sobre la práctica.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	3
Semana 5	5	Aula: Estructura de datos cola. Implementación como TDA sobre una estructura estática. Primitivas coherentes entre implementación estática y dinámica. Aplicaciones de Colas, ejemplos sobre la práctica.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	3

Semana 6	6	Estructura de datos Lista. Implementación dinámica como TDA. Primitivas de inserción: al inicio, al final, en orden, por posición, etc.. Mención de su implementación estática. Aplicaciones de Listas, ejemplos sobre la práctica.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	4
Semana 7	7	Estructura de datos Lista. Implementación dinámica como TDA. Primitivas de eliminación: al inicio, al final, en orden, por posición, etc.. Primitivas de ordenamiento. Aplicaciones de Listas, ejemplos sobre la práctica.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	4
Semana 8	8	Explicación conceptual de listas circulares, implementación de colas y pilas con asignación dinámica de memoria en listas circulares, similitud y diferencia entre las primitivas de pila y cola dinámica en listas circulares. Primitivas coherentes con las ya vistas. Funciones map, filter y reduce.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	4
Semana 9	9	Explicación conceptual de listas doblemente enlazadas. Desarrollo de primitivas para insertar, eliminar y ordenar listas doblemente enlazadas.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	4
Semana 10	10	Aula: Recursividad. Concepto. Elementos básicos de las funciones recursivas. Envoltorios para funciones recursivas. Tipos de recursividad. Desarrollar funciones recursivas de búsqueda, ordenamiento, de cadenas, de listas, etc. Comparar la implementación de funciones recursivas e iterativas. Ventajas y desventajas de implementaciones recursivas vs. iterativas.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	5
Semana 11	11	Estructura de datos Árbol. Árbol binario. Árbol binario de búsqueda. Primitivas. Recorridas. Mención de su compatibilidad con la implementación estática. Funciones para verificar árbol (AVL) semi balanceados, balanceados, completos. Similitud con búsquedas dicotómicas. Comparar implementaciones recursivas vs. iterativas. Presentación de funciones variadas (búsqueda de la clave de ordenamiento, altura, contar hojas, contar no hojas, contar nodos que cumplen una	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	6

		determinada condición, eliminar árbol, eliminar nodos, podar ramas, etc.).			
Semana 12	12	Concepto de índice. Archivo de índice (idx). Implementación de índices sobre un árbol binario de búsqueda. Crear archivo de índice. Regenerar archivo de índice.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	6
Semana 13	13	Parcial 1 Entrega trabajo práctico	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	1 - 2 - 3 - 4 - 5 - 6
Semana 14	14	Devolución parcial 1. Defensa trabajo práctico Ejercitación sobre todos los temas vistos hasta el momento. Consultas sobre el trabajo práctico.	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	1 - 2 - 3 - 4 - 5 - 6
Semana 15	15	Recuperatorio Parcial 1 Entrega trabajo práctico	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	1 - 2 - 3 - 4 - 5 - 6
Semana 16	16	Defensa trabajo práctico Entrega de notas finales	Teoría - Práctica en laboratorio o virtual sincrónico	4 hs	1 - 2 - 3 - 4 - 5 - 6

Evaluación

El proceso de evaluación consta de:

- un examen parcial presencial incluye las unidades 1 a 6.
- un trabajo práctico grupal.
- un examen recuperatorio presencial incluye las unidades 1 a 6.

Las evaluaciones podrán ser teórico - prácticas. o únicamente prácticas o teóricas.

Las evaluaciones teóricas constan de preguntas para desarrollar y/o preguntas del tipo opción múltiple y/o verdadero / falso, justificando la respuesta o no, detectar errores de un código, etc.

Las evaluaciones prácticas constan de la resolución de ejercicios en máquina. Desarrollando el código necesario para resolver el problema solicitado y el lote de pruebas correspondiente.

Una vez entregado el parcial, el mismo es corregido por los docentes, probando que cumpla con la consigna solicitada. A la semana siguiente, se hace una devolución grupal del mismo. Luego se hace la devolución individual.

El trabajo practico grupal se evalúa en diferentes niveles, autoevaluación por parte de los integrantes del grupo; evaluación entre pares, cada grupo verifica el trabajo de otro grupo y hace las correcciones que crea conveniente; por último, los docentes verifican el trabajo original y las correcciones realizadas. Se realiza una puesta en común del ejercicio planteado y se construye una solución grupal.

Primera evaluación	Aproximadamente Semana 13	teórico - práctica	4 hs
Aprobación TP Grupal	A partir de la semana 13	teórico - práctica	4 hs
Recuperatorio	Aproximadamente Semana 15	teórico - práctica	4 hs

Bibliografía obligatoria

Título	Autor	Editorial	Edición	Año
El Lenguaje de Programación C	Kernighan y Ritchie	Prentice Hall	2.ed.	1991
Cómo Programar en C / C++	Deitel y Deitel	Prentice Hall	2a. ed.	c1994
Apuntes de cátedra	Docentes de la cátedra	Disponible en MleL		

Bibliografía complementaria recomendada

Título	Autor	Editorial	Edición	Año
Data Structures and Program Design in C	Kruse, Leung y Tondo	Prentice Hall		
Estructuras de Datos con C y C++	Langsam, Augenstein y Tenenbaum	Prentice Hall		

Otros recursos obligatorios [Videos, enlaces, otros. Incluir una fila por cada recurso]

Nombre

Otros recursos complementarios [Videos, enlaces, otros. Incluir una fila por cada recurso]

Nombre	
https://pythontutor.com/c.html#mode=edit	Permite visualizar en el navegador lo que la computadora está haciendo paso a paso mientras ejecuta un programa.
https://loom.wecode.io	Plataforma para realizar revisión entre pares