

Carrera INGENIERÍA EN INFORMÁTICA		
Asignatura 3652 - Programación Avanzada		
Trayecto de Programación		
Año académico 2023		
Responsable / Jefe de cátedra Dra. Verónica Inés Aubin		
Carga horaria semanal 4hs	Carga horaria total 64hs	Créditos ----
Modalidad: Presencial		
Correlativas anteriores BASES DE DATOS APLICADAS - PARADIGMAS DE PROGRAMACION	Correlativas posteriores GESTION APLICADA AL DESARROLLO DE SOFTWARE II - SEGURIDAD APLICADA Y FORENSIA	
Conocimientos necesarios -----		

Descripción de la asignatura

Esta asignatura introduce los conceptos de eficiencia y coste de un algoritmo y las herramientas matemáticas básicas para analizarlo. Tras un breve repaso de conceptos y técnicas algorítmicas básicas ya conocidas, se estudian nuevas técnicas como el método voraz (greedy method) y la programación dinámica. Cada una de las técnicas de diseño y análisis estudiadas se ilustra con ejemplos concretos. Se estudia la eficiencia de los algoritmos de ordenamiento y búsqueda, así como también de otros algoritmos. Esta asignatura profundiza en la implementación y desarrollo de TDAs. Presenta las siguientes estructuras de datos: montículo, cola de prioridad y grafos. Se presentan diversas implementaciones de las estructuras de datos mencionadas. Se analizan distintos algoritmos de resolución de problemas de aplicaciones reales en grafos.

Metodología de enseñanza

Se utilizan metodologías activas de enseñanza, especialmente aprendizaje basado en problemas (ABP). Se presenta cada núcleo temático, introduciendo las ideas fundamentales con la asistencia de ejemplos reales, lo que motiva a los alumnos y permite relacionar los contenidos de la materia con las herramientas de desarrollo que están acostumbrados a utilizar. Luego, se formalizan las definiciones correspondientes para complementar y organizar los contenidos en un marco teórico establecido.

Los contenidos de la asignatura se presentan de forma iterativa e incremental, de modo que los estudiantes puedan integrar los nuevos conceptos a los anteriores de forma de lograr aprendizajes significativos.

Se motiva a los estudiantes en el uso de foros internos, además de las plataformas MIEl y 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.

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.

A lo largo de los encuentros se irá desarrollando un TP especial con puntos de control intermedios.

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

Objetivos de aprendizaje

A través de esta asignatura, el alumno 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.
- 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:

- Calcular la complejidad computacional de un problema.
- Comparar la eficiencia de distintos algoritmos para resolver un mismo problema y seleccionar el más adecuado
- Conocer e implementar los algoritmos voraces
- Conocer e implementar las estrategias de programación dinámica como alternativa de resolución de problemas
- Resolver problemas reales aplicando algoritmos de grafos (camino mínimo, árbol abarcador de costo mínimo, etc.).
- Conocer e implementar las estructuras de datos presentadas en la materia.
- Analizar la diferencia de rendimiento de los distintos algoritmos de grafos según la implementación utilizada.

Contenidos mínimos

Tipos de datos abstractos avanzados: Colas de prioridad, montículos y Grafos.
Métodos voraces. Programación dinámica. Diseño y Análisis de Algoritmos Complejidad computacional. Algoritmos. Ordenamiento, Búsqueda y Grafos. Aplicación.

Competencias a desarrollar

Genéricas

- 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.
- Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.
- Concepción, diseño y desarrollo de proyectos de ingeniería en sistemas de información / informática.
- Gestión, planificación, ejecución y control de proyectos de ingeniería en sistemas de información / informática.

Específicas

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

Programa analítico

Unidad 1	<p>Análisis de Algoritmos Introducción de algoritmos conocidos. Algoritmos de ordenamientos y sus implantaciones. Comparaciones de los distintos algoritmos en función de sus operaciones básicas. Algoritmos de ordenamiento y búsqueda. Utilización de estructuras estáticas y dinámicas. Estabilidad y sensibilidad a la entrada de un algoritmo de ordenamiento. Algoritmos recursivos. Divide y Vencerás, partición en subproblemas, recombinación de soluciones. Ejemplos: mergesort, quicksort.</p>
Unidad 2	<p>Complejidad Computacional Problemas P vs. NP. Tipos de complejidad. Coste en tiempo y espacio. Caso peor, mejor y medio. Medición del tiempo de ejecución de un algoritmo. Orden de un algoritmo. Velocidad de crecimiento del tiempo de ejecución. Notación asintótica. Cálculo del tiempo de ejecución de un programa. Efectos al aumentar la velocidad del computador y/o al aumentar el tamaño del problema. Análisis del coste de algoritmos iterativos y recursivos.</p>
Unidad 3	<p>Estructura de datos Tipos de datos abstractos: montículos, Colas de prioridad. Operaciones de colas con prioridades. Distintas implementaciones.</p>

Unidad 4	Programación Dinámica Principio de optimalidad. División en subproblemas. Recursividad vs. aproximaciones top-down y bottom-up. Análisis con base en los conceptos de Complejidad Computacional.
Unidad 5	Algoritmos de Grafos Conceptos de Grafos. Teoría. Representaciones: matrices de adyacencia, listas de adyacencia. Búsqueda en profundidad (DFS). Búsqueda en anchura (BFS). Algoritmos de Grafos: Dijkstra, Floyd, Warshall, Prim, Kruskal y coloreo de grafos. Resolución de problemas reales y ficticios.

Planificación de actividades					
Semana	Clase	Actividad	Tipo	Duración estimada	Unidad/des
Semana 1	1	<p>Presentación de los docentes del curso, breve explicación de las pautas generales de la materia. Metodología de trabajo de la materia. Trabajo por pares.</p> <p>Métodos de Ordenamiento Elementales: Burbujeo, Inserción, Selección.</p> <p>Comparaciones de los distintos algoritmos en función de sus operaciones básicas. Estabilidad y sensibilidad al input de un algoritmo de ordenamiento. Caso peor, mejor y medio. Utilización de estructuras estáticas y dinámicas. Algoritmos de búsqueda.</p>	Teoría - Práctica en laboratorio	4 hs	Unidad 1
Semana 2	2	<p>Métodos de Ordenamiento No Elementales: Fusión, QuickSort, ShellSort. Cuenta de Distribución. Algoritmos recursivos. Divide y Vencerás, partición en subproblemas, recombinación de soluciones.</p>	Teoría - Práctica en laboratorio	4 hs	Unidad 1
Semana 3	3	<p>Complejidad Computacional</p> <p>Medición del tiempo de ejecución de un algoritmo. Orden de un</p>	Teoría - Práctica en laboratorio	4 hs	Unidad 2

		algoritmo. Velocidad de crecimiento del tiempo de ejecución. Función O.			
Semana 4	4	Complejidad Computacional. Efectos al aumentar la velocidad del computador y/o al aumentar el tamaño del problema. Regla de la suma y del producto Práctica	Teoría - Práctica en laboratorio	4 hs	Unidad 2
Semana 5	5	TDA Montículos - Cola de Prioridad	Teoría - Práctica en laboratorio	4 hs	Unidad 3
Semana 6	6	Algoritmos de fuerza bruta, voraz. Programación Dinámica: Principio de optimalidad. División en subproblemas. Recursividad vs. aproximaciones top-down y bottom-up	Teoría - Práctica en laboratorio	4 hs	Unidad 4
Semana 7	7	Programación Dinámica. Práctica y análisis de resoluciones.	Teoría - Práctica en laboratorio	4 hs	Unidad 4
Semana 8	8	Introducción a Grafos. Teoría de grafos. Nodo. Arista. Grado. Conceptos constitutivos. Caminos y ciclos de Hamilton y Euler. Isomorfismo. Representaciones: matrices de adyacencia, listas de adyacencia, lista de incidencia. Búsqueda en profundidad (DFS). Búsqueda en anchura (BFS).	Teoría - Práctica en laboratorio	4 hs	Unidad 5
Semana 9	9	Algoritmos de Grafos. Prim y Kruskal. Resolución de ejercicios	Teoría - Práctica en laboratorio	4 hs	Unidad 5
Semana 10	10	Algoritmos de Grafos. Dijkstra, Floyd, Warshall. Resolución de ejercicios	Teoría - Práctica en laboratorio	4 hs	Unidad 5

Semana 11	11	Algoritmos de Grafos. Coloreo de grafos. Secuencial, Welsh-Powell y Matula. Resolución de ejercicios	Teoría - Práctica en laboratorio	4 hs	Unidad 5
Semana 12	12	Resolución de ejercicios con Grafos, estilo parcial con OIA nivel 3. Entrega trabajo práctico	Teoría - Práctica en laboratorio	4 hs	Unidad 5
Semana 13	13	Parcial	Parcial en laboratorio	4 hs	Unidades 1, 2, 3, 4 y 5
Semana 14	14	Algoritmos de Grafos. Algoritmos adicionales de interés (como flujo máximo o algoritmo de Ford, A*)	Teoría - Práctica en laboratorio	4 hs	Unidad 5
Semana 15	15	Recuperatorio Parcial Reentrega trabajo práctico	Recuperatorio en laboratorio	4 hs	Unidades 1, 2, 3, 4 y 5
Semana 16	16	Defensa trabajo práctico. Entrega de notas finales	Defensa TP en laboratorio	4 hs	Unidades 1, 2, 3, 4 y 5

Evaluación

El proceso de evaluación consta de:

- Un examen parcial presencial
- Un trabajo práctico grupal. La aprobación del trabajo práctico conlleva la aprobación del código entregado y una defensa oral individual del mismo.
- Un examen recuperatorio presencial.
- Un trabajo práctico grupal.

Las evaluaciones (examen parcial y recuperatorio) serán teórico-prácticas.

Constan de preguntas para desarrollar y/o del tipo opción múltiple justificando la respuesta y un porcentaje no menor al 50% de ejercicios prácticos donde el alumno deberá resolver problemas desarrollando código con las pruebas correspondientes. En examen parcial se evalúan los contenidos de la Unidad 1 a la Unidad 5. Los parciales corregidos serán entregados en mano a los estudiantes, teniendo la posibilidad de realizar preguntas sobre las correcciones efectuadas. Previamente a la entrega de los parciales corregidos, se explicará brevemente la solución a los problemas planteados haciendo hincapié en los puntos donde se observaron los errores más comunes. Una vez completado este procedimiento las calificaciones son confirmadas y volcadas al sistema Guaraní.

El trabajo práctico será sobre los contenidos vistos en las Unidades 2y 4

El trabajo se realizará en forma grupal, con equipos de 4 a 6 personas. La aprobación del trabajo práctico conlleva la aprobación del código entregado y una defensa oral individual del mismo.

Primera evaluación	Semana 13	Teórico-práctica	4 hs
Segunda evaluación	A partir de la semana 13	Trabajo práctico	4 hs
Recuperatorio	Semana 15	Teórico-práctica	4 hs

Bibliografía obligatoria				
Título	Autor	Editorial	Edición	Año
Algoritmos en C++	Sedgewick, Robert	Addison-Wesley Iberoamericana		1995
Estructuras de datos y algoritmos	Aho, Hopcraft, Ullman	Addison Wesley Longman		1998
Estructuras de matemáticas discretas para la computación	Kolman, Bernard	Prentice-Hall Hispanoamérica		1997
Estructuras de datos y algoritmos	Mark Allen Weiss	Addison Wesley Iberoamericana		1995

Bibliografía complementaria recomendada				
Título	Autor	Editorial	Edición	Año
Grokking algorithms	Aditya Bhargava	Manning	Primera	2016
Code	Charles Petzhold	Microsoft Press	Segunda	2022
Introduction to Graph Theory	Richard Trudeau	Dover	-	1994

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	