

Asignatura: **Algoritmos y Estructuras de Datos**

Código: 10-09802	RTF	7
Semestre: 2	Carga Horaria	96
Bloque: Ciencias Básicas	Horas de Práctica	48

Departamento: Computación

Correlativas:

- Estructuras Discretas
- Fundamentos de Programación

Contenido Sintético:

- Historia y visión general
- Herramientas relevantes, estándares y/o restricciones de ingeniería
- Estructuras de datos lineales y no lineales
- Análisis de algoritmos básicos
- Estrategias algorítmicas
- Algoritmos clásicos para tareas comunes de ordenamiento y búsqueda
- Análisis y diseño de algoritmos de aplicación específicos
- Complejidad algorítmica

Competencias Genéricas:

- CG1: Identificar, formular y resolver problemas de ingeniería. Media
- CG4: Utilizar de manera efectiva las técnicas y herramientas de aplicación en ingeniería. Alta

Aprobado por HCD: 1042-HCD-2023

RES: Fecha: 27/11/2023

Competencias Específicas:

CE1.3 Conocer, desarrollar nuevos e implementar algoritmos y estructuras de datos. Alta

Presentación

Los algoritmos y las estructuras de datos son esenciales a la computación, ambos forman parte de los conceptos centrales en que esta ciencia se basa. Las estructuras de datos han sido desarrolladas para representar adecuadamente la información de ciertos tipos de problemas de procesamiento de datos, de forma tal de poder implementar adecuadamente las soluciones de estos. Los algoritmos realizan el procesamiento sobre los datos, por lo que la combinación adecuada de algoritmos con estructuras permiten desarrollar soluciones informáticas eficientes.

La asignatura comprende la presentación y usos de las estructuras de datos y algoritmos de uso frecuente y generalizado y están destinados a que el alumno adquiera conocimientos y capacidades su aplicación. Además es teniendo en cuenta el uso eficiente tanto de la memoria como de las unidades de procesamiento para el abordaje de las distintas situaciones en las cuales se utilizan dichos algoritmos y estructuras en los sistemas de computación.

El desarrollo de los contenidos es gradual en su complejidad para facilitar el aprendizaje de los mismos y en toda oportunidad se presentan acompañados de su uso concreto de forma tal que los fundamentos de estos puedan contrastarse inmediatamente con su utilización.

Es tenida en cuenta la ubicación de la asignatura dentro del trayecto formativo de los alumnos en cuanto a los requisitos previos para poder tomar el curso adecuadamente y el formato de presentación de los temas. Además, en la preparación y capacitación del alumnado se tienen en cuenta las competencias necesarias para poder abordar las asignaturas posteriores a la presente a lo largo del plan de estudios de la carrera de Ingeniería en Computación.

Contenidos

Unidad 1: Punteros y memorias

- 1.1 Variables, memoria y direcciones
- 1.2 Paso de parámetros mediate punteros
- 1.3 Uso dinámico de la memoria. Reserva y liberación de memoria
- 1.4 Aritmética de Punteros
- 1.5 Usos y Ejemplos

Unidad 2: Conceptos básicos de objetos y clases

- 2.1 Clases y Objetos
- 2.2 Métodos y atributos privados y públicos
- 2.3 Herencia
- 2.4 Constructores
- 2.5 usos y ejemplos

Unidad 3: Estructuras lineales

- 3.1 Arreglos y Vectores
- 3.2 Listas simplemente enlazadas
- 3.3 Pilas
- 3.4 Colas
- 3.5 Usos y Ejemplos

Unidad 4: Complejidad algorítmica

- 4.1 Complejidad temporal y espacial
- 4.2 Funciones asintóticas Notación Big O, Omega O y Theta O

4.3 Ejemplos

Unidad 5: Algoritmos de ordenamiento y búsqueda

- 5.1 Algoritmos de ordenamiento de Orden cuadrático
- 5.2 Algoritmos de ordenamiento de Orden logarítmico
- 5.3 Comparación y usos apropiados de cada algoritmo
- 5.4 Algoritmos de búsqueda: lineal, binaria y otros
- 5.5 Usos y ejemplos

Unidad 6: Estructuras tipo árbol

- 6.1 Definición y formas de implementar árboles.
- 6.2 Árbol binario y m-ario
- 6.3 Balances y recorridos de arboles binarios
- 6.4 Otros tipos de árboles usuales
- 6.5 Usos y ejemplos

Unidad 7: Otras estructuras de datos

- 7.1 Tablas y funciones de dispersión
- 7.2 Conjuntos y bolsas (set y bags)
- 7.3 Mapas de Bits

Unidad 8: Grafos

- 8.1 Definición y formas de implementar grafos
- 8.2 Algoritmos sobre grafos dirigidos
- 8.3 Algoritmos sobre grafos no dirigidos
- 8.4 Usos ejemplos

Unidad 9: Estrategias algorítmicas

- 9.1 Algoritmos "Divide y Vencerás"
- 9.2 Algoritmos voraces
- 9.3 Algoritmos de *backtracking*

Unidad 10: Visión de conjunto e historia de los algoritmos y las estructuras de datos

- 10.1 Revisión histórica de los algoritmos y las estructuras de datos

Metodología de enseñanza

Los contenidos de la asignatura serán impartidos bajo una metodología de clases teórico - prácticas con la activa participación del alumnado en las mismas. La asignatura tiene una impronta muy fuerte en el diseño y elaboración de programas, en cuya elaboración los contenidos teóricos son necesarios para la resolución de los mismos pero la importancia de su conocimiento es efectivamente adquirida en oportunidad del desarrollo de los programas, por lo que la combinación de conceptos teóricos con programas que los implementen genera un alto impacto en la capacidades que los alumnos adquieren.

Las clases son impartidas combinando la presentación del tema y su desarrollo junto con la exposición de programas concretos que los implementen, de forma tal que el alumno reconoce inmediatamente el impacto de la aplicación de los conceptos en el desarrollo de

algoritmos, su ventajas, alcances y limitaciones. Dicho esto, en toda oportunidad está presente un entorno de desarrollo y ejecución de programas que determina la condición de una metodología teórico-práctica.

Evaluación

La metodología de la evaluación es por medio del método de la resolución de problemas informáticos y el desafío de diseñar e implementar la solución del mismo. A lo largo del curso se presentan tres proyectos a desarrollar bajo la metodología precedente, donde para su solución se deben aplicar conceptos impartidos en clases. La solución de los mismos puede ser encarada con enfoques diferentes por lo que el alumnado tiene que desarrollar habilidades para comprender el problema y solucionarlo basado en sus conocimientos adquiridos.

A los efectos de evaluar el proceso de aprendizaje del alumno, no solo se tendrá en cuenta la aplicación del algoritmo y la estructura de datos utilizada, sino que también la forma de haber abordado la solución del problema, la justificación de la técnica utilizada y la claridad a la hora de comunicar todo el proceso constructivo del proyecto

Si bien para el desarrollo de los proyectos pueden utilizarse diversos lenguajes de programación, se solicita que en la resolución sea utilizado un lenguaje de programación compilado y usual en el desarrollo de programación de sistemas embebidos para que sea parte de la formación integral del alumno en su carrera.

Condiciones de aprobación

Se presentan tres(3) proyectos a desarrollar a lo largo del curso. Para su ejecución se solicita trabajo en equipo y que la presentación y defensa de los mismos se realice en forma conjunta por parte de los miembros del equipo frente al docente. También se solicita un documento escrito que exponga el diseño y la solución desarrollada.

Para alcanzar la regularidad, deben aprobarse dos(2) de los tres proyectos y asistencia a clases del 80%. La promoción se logra aprobando los tres proyectos.

La nota final en caso de promoción es el promedio de las tres evaluaciones. Para el caso del alumno regular, la nota es el promedio ponderado de las dos evaluaciones aprobadas durante el cursado, a un 25% cada una de estas, y la nota del examen final ponderado al 50%. Al alumno libre le corresponde la nota del examen libre.

Actividades prácticas y de laboratorio

Al ser la metodología de clases teóricas - prácticas, el alumno se acompaña en las mismas por computadoras donde desarrollan los ejemplos planteados y las actividades prescriptas.

Desagregado de competencias y resultados de aprendizaje

Al aprobar el curso el alumno adquiere una serie de habilidades y conocimientos técnicos propios de la carrera como también habilidades blandas. Dentro de los primeros podemos citar la identificación de problemas y diseño y desarrollo de soluciones propios de la Ingeniería en Computación, la aplicación de algoritmos y estructuras de datos de uso frecuente en computación, el diseño, desarrollo y depuración de programas y competencias de adquisición y aplicación de nuevos conceptos. Uso de lenguajes de programación compilables. Dentro de las habilidades genéricas adquiridas podemos citar la capacidad de trabajo en equipo y la documentación y comunicación de los trabajos técnicos realizados. Estos resultados de aprendizaje se evaluarán en las instancias de evaluación previstas mediante el uso de rúbricas diseñadas al efecto.

Bibliografía

Principal

Estructuras de datos con C++. Objetos, abstracciones y diseño, E. Koffman & P.Wolfgang Introduction to Algorithms (4th Edition), T. Cormen, C. Leiserson, R. Rivest,& C.Stein

Adicional

Algorithms in C++, Robert Sedgewick & Kevin Wayne Algorithms and Data Structures, Niklaus

Wirth

The Design and Analysis of Computer Algorithms (3rd Ed.), A. Aho, J. Hopcroft, J. Ullman

