FCEFyN
FACULTAD DE CIENCIAS EXACTAS, FÍSICAS y NATURALES

UNC Universidad Nacional de Córdoba

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: RES: Fecha:

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 mediates 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

Esta asignatura se encuentra en el plan de estudios de la carrera de Ingeniería en Computación en el 2do semestre de primer año con una carga horaria de 96 horas.
La misma desarrolla los temas de estructuras de datos lineales (listas, listas y colas), no lineales (árboles, grafos y otras) y en el aprendizaje, optimización, validación e implementación de algoritmos (búsqueda, ordenamiento, etc.) sobre esas estructuras de datos, con una introducción en los temas de punteros y clases necesarios. Esta asignatura se articula con la asignatura anterior de Fundamentos de la Programación. Los contenidos se desarrollan de manera teórica a través de la explicación conceptual de los temas y su justificación en términos de optimización. La parte práctica se desarrolla con la participación de los estudiantes en la implementación de las estructuras y los algoritmos en lenguaje C++ directamente sobre las computadoras, mediante trabajos grupales e individuales.

La asignatura tiene una impronta muy fuerte en el diseño e implementación 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 y prueba de los programas, logrando que los mismos funcionen correctamente, por lo que la combinación de conceptos teóricos con programas que los implementen genera en los alumnos las habilidades necesarias que requieren para la continuidad de la carrera y profesionalmente en cuanto a la programación.

Las clases teóricas 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, sus ventajas, alcances, optimización y limitaciones. En las prácticas los alumnos aplican estos conocimientos de manera grupal e individual logrando efectivamente que programas y algoritmos desarrollados por ellos funcionen adecuadamente. En toda oportunidad está presente un entorno de desarrollo y ejecución de los programas. En ese sentido, todas las actividades se enmarcan en la utilización de software de código abierto y gratuito para la resolución de ejercicios y problemas contextualizados, en forma individual o grupal.

- Los contenidos teóricos, se imparten de manera remota en tanto se ajustan adecuadamente a las estrategias de enseñanza-aprendizaje establecidas en la materia; y brindan mayor flexibilidad de asistencia a los estudiantes.Los estudiantes podrán participar de las clases mediadas por tecnología, desde ubicaciones externas o utilizando las instalaciones y la conectividad disponible en la Facultad.
- Para las instancias prácticas, ofrecerán comisiones de dictado tanto en modalidad presencial físico y presencial remoto, que son acordes a la metodología de enseñanza-aprendizaje de los contenidos teóricos y prácticos de la materia y brindan alternativas flexibles para los estudiantes de primer año. En el caso de las clases presenciales, las mismas se desarrollarán en los Laboratorios de Informática de la Facultad.

Las instancias de presencialidad remota se desarrollarán en el marco de las pautas institucionales vigentes. Se utilizará la plataforma Google Meet (o la herramienta que la reemplace) para posibilitar la interacción y el trabajo colaborativo. El acceso será restringido y requiere la identificación obligatoria del estudiante a través de su cuenta institucional (IdUNC).

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, asimismo como demostrar los conocimientos teóricos requeridos.

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.

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

La evaluación se llevará a cabo mediante dos (2) exámenes parciales durante el cuatrimestre de cursado, con la posibilidad de recuperar uno (1) de ellos (por ausencia o aplazo, tanto para regularizar la materia como para alcanzar la promoción), y la realización de trabajos prácticos de laboratorios, según la programación de la cátedra. Cada parcial constará de una evaluación de corrección automatizada incluyendo el desarrollo y validación de programas y respuestas a preguntas sobre el contenido práctico. El parcial para aquellos que alcancen la aprobación de las pruebas automatizadas finalizará con una entrevista personal con el docente a cargo de la evaluación donde se evaluarán los conocimientos, las técnicas utilizadas, su pertinencia o no .
El docente a cargo evaluará el desempeño y desarrollo de las competencias de acuerdo con la rúbrica que se detalla más abajo. La instancia de evaluación se aprueba cumplimentando el 60% de la exigencia de cada caso.

En todos los casos el examen automatizado y las entrevistas se realizarán de manera individual y presencial física.

Para alcanzar la regularidad, debe aprobarse un parcial (1), el 80% de los trabajos prácticos que se propongan y la asistencia a clases del 80%. La promoción se logra aprobando los dos parciales, además de las condiciones de regularidad.

La nota final en caso de promoción es el promedio de los dos parciales.

Desagregado de competencias y resultados de aprendizaje

Al aprobar el curso el alumno adquiere una serie de habilidades y conocimientos técnicos específicos de la carrera como también competencias genéricas. En cuanto a la competencia específica se describe como “Conocer, desarrollar nuevos e implementar algoritmos y estructuras de datos.” (Alta)

Dentro de las habilidades genéricas adquiridas se requieren, según el plan de estudios: “Competencia para identificar, formular y resolver problemas de ingeniería” (medio) y “Competencia para utilizar de manera efectiva las técnicas y herramientas de la ingeniería” (alto).

A continuación se presenta el desagregado de las competencias, los resultados de aprendizaje y los niveles de competencia.

Indicadores	Nivel			
	Muy Alto	Alto	Medio	Bajo

CE1.3: Conocer, desarrollar nuevos e implementar algoritmos y estructuras de datos.	RA1 (Selección): Selecciona la estructura y el algoritmo óptimo basándose en un análisis riguroso de complejidad (O). RA2 (Implementación): Codifica sin errores lógicos, manejando casos borde y gestión de memoria. Código limpio y modular. RA3 (Adaptación): Adapta algoritmos creativamente para resolver problemas inéditos de forma escalable. RA4 (Teoría): Fundamenta la solución mediante teoremas y propiedades matemáticas (invariantes, recurrencia). Vincula el modelo abstracto con la implementación física.	RA1 (Selección): Elige una estructura adecuada y realiza un análisis de complejidad mayormente correcto. RA2 (Implementación): Implementa correctamente con errores menores. El código es funcional y legible. RA3 (Adaptación): Adapta algoritmos conocidos a nuevos contextos correctamente. RA4 (Teoría): Explica el funcionamiento interno de la estructura usando terminología técnica precisa. Demuestra comprensión de las propiedades básicas.	RA1 (Selección): Elige estructuras genéricas o sub-óptimas (fuerza bruta). El análisis de complejidad es débil o inexistente. RA2 (Implementación): El código funciona en casos ideales pero falla en casos borde. Estilo desordenado. RA3 (Adaptación): Intenta adaptar soluciones pero el resultado es confuso o poco eficiente. RA4 (Teoría): Relaciona superficialmente la teoría. Describe "qué" hace el algoritmo pero le cuesta explicar "cómo" o "por qué" funciona.	RA1 (Selección): Selecciona estructuras incoherentes con el problema. Desconoce el coste computacional. RA2 (Implementación): El código no compila o tiene errores graves. No hay gestión de recursos. RA3 (Adaptación): No logra desvincularse de ejemplos de libro; copia sin adaptar o no entrega solución. RA4 (Teoría): No vincula los conceptos estudiados. Repite definiciones de memoria sin aplicación al contexto o confunde términos básicos.
CG1 Competencia para identificar, formular y resolver problemas de ingeniería	RA1 Relaciona el texto entregado con los contenidos teóricos estudiados y la actividad a desarrollar. RA1 Compara las ideas y conceptos del texto. RA2 Reconoce toda la información explícita e infiere las principales informaciones implícitas. RA2 Determina los elementos faltantes para la realización de una actividad. RA3 Comunica los resultados en un lenguaje comprensible y usando la notación que corresponde.	RA1 Relaciona parcialmente el texto entregado con los contenidos teóricos estudiados y la actividad a desarrollar. RA1 Compara parcialmente las ideas y conceptos del texto. Reconoce casi toda la información explícita e infiere las principales informaciones implícitas. RA2 Determina los elementos faltantes para la realización de una actividad. RA3 Comunica los resultados en un lenguaje comprensible y usando la notación que corresponde.	RA1 Relaciona escasamente el texto entregado con los contenidos teóricos estudiados y la actividad a desarrollar. RA1 Compara escasamente las ideas y conceptos del texto. Escasamente reconoce toda la información explícita e infiere las principales informaciones implícitas. RA2 No determina los elementos faltantes para la realización de una actividad. RA3 Comunica los resultados sin la notación ni las unidades que corresponden.	RA1 No relaciona el texto entregado con los contenidos teóricos estudiados y la actividad a desarrollar RA1 No compara las ideas y conceptos del texto. RA2 No reconoce la información explícita y no infiere las principales informaciones implícitas. RA2 No determina los elementos faltantes para la realización de una actividad. RA3 Comunica los resultados sin coherencia en el valor y las unidades pertinentes.
CG4 Competencia para utilizar de manera efectiva las	RA1 Planifica e implementa estrategias de trabajo. RA1 Identifica los elementos comunes pertinentes.	RA1 Implementa estrategias de trabajo. RA1 Identifica algunos elementos comunes pertinentes.	RA1 Implementa estrategias de trabajo. RA1 Reconoce algún elemento común.	RA1 Copia estrategias de trabajo. RA1 No hay elementos comunes pertinentes.

técnicas y herramientas de la ingeniería	RA1 Realiza un borrador del texto, utilizando listados, esquemas y cuadros. RA2 Explicita un adecuado marco conceptual. RA2 Utiliza infografía y representaciones adecuadas. RA3 Fundamenta el resultado en forma verbal, oral o escrita. RA3 Verifica que la solución coincide con las predicciones. RA3 En caso de obtener incoherencia, rechaza el resultado y revisa todo el procedimiento.	RA1 Realiza un borrador del texto, utilizando esquemas y cuadros. RA2 Explicita un marco conceptual. RA2 Utiliza infografía y representaciones adecuadas. RA3 Fundamenta el resultado en forma escrita. RA3 Verifica que la solución coincide con las predicciones. RA3 En caso de obtener incoherencia, justifica el resultado y revisa parte del procedimiento.	RA1 Realiza un borrador, utilizando cálculos. RA2 Explicita un escaso marco conceptual. RA2 No utiliza infografía y representaciones adecuadas. RA3 No se fundamenta el resultado. RA3 Verifica que la solución coincide con las predicciones, pero no determina, ni revisa el procedimiento.	RA1 No realiza borrador de esquemas o cálculos. RA2 Nulo marco conceptual. RA2 No utiliza y representaciones RA3 No justifica los resultados. RA3 No verifica si la solución coincide con las predicciones. RA3 No determina incoherencias

Bibliografía

Principal

- Koffman E., Wolfgang P. (2008). Estructuras de datos con C++. Objetos, abstracciones y diseño, McGraw-Hill
- Cormen T, Leiserson C., Rivest R., Stein C. (2022). Introduction to Algorithms (4th Edition), MIT Press

Complementaria

- Sedgewick R., Wayne K. (2011). Algorithms in C++, Princenton University
- Wirth N (1985). Algorithms and Data Structures, Prentice Hall
- Aho A., Hopcroft J., Ullman J. (2004). The Design and Analysis of Computer Algorithms (3rd Ed.), Pearson Education