

UNIDAD DE APRENDIZAJE

IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE

Unidad académica: Facultad de Contaduría Administración e Informática							
Programa educativo: Maestría en Optimización y Cómputo Aplicado							
Unidad de aprendizaje: Laboratorio de Programación				Ciclo de formación: Especializado			
				Eje de formación: Disciplinar			
				Semestre: 1 al 4			
Elaborado por: Dra. Lorena Díaz González				Fecha de elaboración: 1 diciembre 2022			
Actualizado por: M en E. Ana Linda Pineda Mendez y Dra. Lorena Díaz González				Fecha de revisión y actualización: 1 diciembre 2022			
Clave:	Horas teóricas:	Horas prácticas:	Horas totales:	Créditos:	Tipo de unidad de aprendizaje:	Carácter de la unidad de aprendizaje	Modalidad
No Aplica	2	2	4	6	Optativa	Posgrado	presencial
Programa (s) educativo (s) en los que se imparte: Maestría en Optimización y Cómputo Aplicado							

PRESENTACIÓN

Se presentan los conceptos de algoritmos de ordenamiento y estructuras de datos dinámicas, así como su implementación en el lenguaje C o C++.

El estudiante comprenderá los fundamentos de la POO y los implementará en un lenguaje de su elección que soporte la POO, por ejemplo Java o C++.

Este curso es una parte importante en los procesos de abstracción que los estudiantes requieren para comprender la teoría de la computación y la programación. Asimismo, proporciona las herramientas útiles para la programación en cualquier lenguaje.

PROPÓSITOS

El desarrollo de los programas y algoritmos es fundamental en ciencias y en ingeniería. El comportamiento práctico de un software depende de los algoritmos usados en este software y la eficiencia de estos algoritmos (el tiempo que van a tardar estos algoritmos en la práctica). Por esta razón, es crucial el diseño de los programas y algoritmos eficientes para todo tipo de software.

El objetivo del curso es introducir al estudiante al análisis y diseño de algoritmos eficientes usando la programación modular o estructurada. Asimismo, este curso se propone introducir al estudiante al proceso de abstracción utilizando la Programación Orientado a Objetos (POO). El estudiante será

capaz de llevar un problema del mundo real a un modelo de computadora concentrándose en lo relevante dejando de lado los detalles de su implementación.

COMPETENCIAS QUE CONTRIBUYEN AL PERFIL DE EGRESO

Competencias básicas
a. Lectura, análisis y síntesis b. Comunicación oral y escrita c. Aprendizaje estratégico d. Razonamiento lógico-matemático e. Razonamiento científico
Competencias genéricas
Competencias genéricas: a. Cognitivas-metacognitivas • Resolución de problemas • Pensamiento crítico • Creatividad b. Socioemocionales genéricas • Orientación al logro • Apertura a la experiencia c. Digitales genéricas • Búsqueda, valoración y gestión de información • Comunicación y colaboración en línea • Resolución de problemas técnicos d. Socioculturales genéricas • Comunicación en segundo idioma • Responsabilidad social y ciudadana
Competencias laborales
Competencias específicas disciplinares

- a) Conocer, comprender y aplicar conceptos básicos de matemática discreta, teorías, análisis y diseño de algoritmos y métodos estadísticos, mediante el estudio de modelos teóricos de la computación para generar soluciones de cómputo e informáticas
- b) Plantea el modelo matemático de problemas de optimización y cómputo aplicado para economizar recursos en el desarrollo sostenible y sustentable mediante la aplicación del método científico.
- c) Analiza metodologías para producir nuevo conocimiento a través de proyectos de investigación

Competencias transferibles para el trabajo

Digitales para el trabajo

- Aplica herramientas de análisis de datos para generar información que auxilie en la toma de decisiones utilizando repositorios de datos públicos y/o privados.

Competencias para el trabajo transdisciplinar

- Desarrolla investigación en equipo, a fin de identificar puentes de interconexión entre disciplinas que aporten nuevo conocimiento a través de discusiones y búsqueda de información compartida

Competencias para el aprendizaje a lo largo de la vida laboral (aprender, reaprender y desaprender)

- Desarrolla conocimientos, competencias, actitudes y valores necesarios para tener la capacidad de adaptación a nuevas demandas profesionales que surjan en el mercado laboral, mediante un aprendizaje flexible personalizado por cada estudiante definido por el asesor de tesis con apoyo del paradigma de la educación 4.0

CONTENIDOS

Bloques	Temas
Módulo 1: Algoritmos de ordenamiento	1.1 Algoritmo de ordenamiento de las burbujas: implementación iterativa y recursiva de este algoritmo en C. 1.2 Algoritmos más eficientes de ordenamiento: MergeSort, QuickSort y sus implementaciones recursivas en C. 1.3 Métodos divide y vencerás para el diseño de algoritmos. 1.4 Algoritmo de Euclides y su implementación recursiva en C.
Módulo 2: Estructuras dinámicas de datos	2.1 Definición de tipos de datos abstractos.

Bloques	Temas
Temas y subtemas	2.2 Colas, pilas, listas simple y doblemente enlazadas, y sus implementaciones en C con apuntadores. 2.3 Grafos y árboles, y su descripción con apuntadores. 2.3 Árboles binarios de búsqueda. Recorridos: Postorder, Inorder y Preorder. 2.4 Búsqueda profundidad primero y altura primero. 2.5 Implementaciones recursivas de estas estructuras de datos. 2.6 Cobertura mínima de árboles ("minimal spanning trees"): algoritmos de Prim y Kruskal.
Módulo 3: Programación orientada a objetos Temas y subtemas	3.1 Conceptos: clases, objetos, métodos y variables de instancia. 3.2 Métodos: Importancia de los métodos establecer/obtener; Tipos de acceso a métodos; Métodos y campos estáticos; Paso de argumentos a métodos; Métodos con múltiples parámetros; Sobrecarga de métodos. 3.3 Clases y objetos: Control de acceso a los miembros; Constructores predeterminados; Constructores sobrecargados; Enumeraciones; Miembros de clase estáticos; Recolección automática de basura; Reutilización de software; Abstracción de datos y encapsulamiento; Paquetes de clases; Instancias: creación y manejo de instancias. 3.4 Manejo de excepciones: Generalidades acerca del manejo de excepciones y jerarquía de excepciones.
Módulo 4: Conceptos avanzados de programación orientada a objetos	Herencia: 4.1 Superclases y subclases. 4.2 Relaciones entre superclases y subclases. 4.3 Constructores en las subclases. Polimorfismo: 4.4 Métodos abstractos. 4.5 Interfaces. 4.6 Clases abstractas e Herencia múltiple.

ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE

Estrategias de aprendizaje sugeridas (Marque con X)		
Aprendizaje basado en problemas	☒	Nemotecnia
Estudios de caso	☒	Análisis de textos

Trabajo colaborativo	☒	Seminarios	☐
Plenaria	☐	Debate	☐
Ensayo	☐	Taller	☐
Mapas conceptuales	☐	Ponencia científica	☒
Diseño de proyectos	☒	Elaboración de síntesis	☐
Mapa mental	☐	Monografía	☐
Práctica reflexiva	☐	Reporte de lectura	☒
Trípticos	☐	Explosión oral	☐
Otros			
Estrategias de enseñanza sugeridas (Marque X)			
Presentación oral (conferencia o exposición) por parte del docente	☒	Experimentación (prácticas)	☒
Debate o Panel	☐	Trabajos de investigación documental	☐
Lectura comentada	☒	Anteproyectos de investigación	☐
Seminario de investigación	☐	Discusión guiada	☐
Estudio de Casos	☒	Organizadores gráficos (Diagramas, etc.)	☐
Foro	☐	Actividad focal	☐
Demostraciones	☐	Analogías	☐
Ejercicios prácticos (series de problemas)	☒	Métodos de proyectos	☒
Interacción con la realidad (a través de videos, fotografías, dibujos software especialmente diseñado)	☒	Exploración de la web	☒
Archivo	☒	Portafolio de evidencias	☐
Ambiente virtual (foros, chat, correos, ligas a otros sitios web, otros)	☐	Enunciado de objetivo o intenciones	☐
Otra, especifique (lluvia de ideas, mesa redonda, textos programados, cine, teatro, juego de roles, experiencia estructurada, diario reflexivo, entre otras): mesa redonda			

CRITERIOS DE EVALUACIÓN

Criterios	Porcentaje
- Exámenes teóricos y programas para implementar la teoría de POO.	20%
- Proyecto final que incluya conocimientos relacionados con su tema de tesis	20%
- Participación en clases	20%
- Exposición de seminarios por los alumno	20%
- Tareas prácticas	20%
Total	100%

PERFIL DEL PROFESOR

Doctor con experiencia en el área de programación estructurada y orientada a objetos.

REFERENCIAS BIBLIOGRÁFICAS

Básicas:

- Tenenbaum, A. M., Langsam, Y. y Augenstein, M. Data structures in C. Ed. Prentice Hall.
- Aho, Hopcroft y Ullman. 1988. Estructura de datos y algoritmos. Ed. Addison Wesley Iberoamericana.H. Deitel y P. Deitel. Cómo programar en Java, 9a edición, Pearson (2012).

Complementarias:

- Weiss, Mark Allen. 1995. Estructuras de datos y algoritmos. Ed. Addison-Wesley.
- F. Gutiérrez, F. Duran, y E. Pimentel. Programación Orientada a Objetos con Java, Editorial Paranifo, S.A. (2007)
- David J. Barnes, Michael Kolling, Programación Orientada a Objetos con java usando bluej, 5ª Edición, Editorial Pearson (2013)

Web:-----

Otros: -----