

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: Principios y Paradigmas 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

Durante este curso se definirán los principios de lenguajes de programación y se establecerán las diferencias entre los paradigmas de programación estructural, funcional, lógica y orientada a objetos. Particularmente, el estudiante será capaz:

- (i) Distinguir diferentes tipos de lenguajes de programación a partir de sus orígenes conceptuales;
- (ii) Reconocer las ventajas y desventajas de los diferentes tipos de lenguajes de programación, con la finalidad de buscar la solución a un problema en particular;
- (iii) Comprender los conceptos modernos de las nuevas generaciones de lenguajes de programación tales como clases, métodos, objetos, herencia, polimorfismo, paquetes genéricos, manejo de excepciones, entre otros; y,
- (iv) Generalizar los principios y orígenes de la programación funcional y lógica.
- (v) Identificar los problemas en los cuales es conveniente aplicar la concurrencia y programación paralela.

PROPÓSITOS

Implementar algoritmos y comprender los conceptos teóricos de los lenguajes de programación, tales como analizador léxico, sintáctico y semántico; funcionamiento de variables y constantes, datos y tipos de datos, expresiones y sentencias, manejo de excepciones y procedimientos, y tipos de datos abstractos y módulos. Así también este curso, tiene el objetivo de presentar los diferentes tipos de paradigmas de programación

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: Introducción	1.1 Panorama histórico. 1.2 Criterios para el diseño de lenguajes.
Módulo 2: Sintaxis y semántica	2.1 Descripción sintáctica de los lenguajes de programación. 2.2 Gramáticas independientes de contexto y BNF. 2.3 Diagramas y grafos sintácticos. 2.4 Técnicas y herramientas para análisis sintáctico. 2.5 Semántica formal
Módulo 3: Parte I: Conceptos fundamentales de lenguajes de programación	3.1 Variables y constantes

Bloques	Temas
	Atributos: nombre dirección, alias, tipo valor. Espacio y alcance. Concepto de ligadura. Análisis de tipo. Compatibilidad de tipos. Alcance y vida de las variables. Ambiente de referencia. Inicialización de variables. 3.2 Datos y tipos de datos Tipos de datos primitivos y compuestos. Tipos de datos definidos por el usuario. Apuntadores y manejo de memoria dinámico. Conversión y equivalencia de tipos. Polimorfismo explícito. 3.3 Expresiones y sentencias Expresiones aritméticas, relacionales y lógicas. Sobrecarga de operadores. La controversia GOTO. Manejo de excepciones.
Módulo 4: Parte II: Conceptos fundamentales de lenguajes de programación	4.1 Procedimientos Semántica de procedimiento. Mecanismos de paso de parámetros, por valor, por copia, por referencia, etc. Sobrecarga de nombres. 4.2 Tipos de datos abstractos Especificación algebraica de tipos de datos abstractos. Mecanismos de tipos de datos abstractos: paquetes en Java y Namespaces C++. Paquetes en Ada. Módulos en ML.
Módulo 5: Paradigmas de lenguajes de programación	5.1 Programación orientada a objetos Conceptos fundamentales, clases, métodos, objetos. Herencia, polimorfismo. Manejo de memoria dinámico y recolección de basura. Lenguajes orientados a objetos. Aplicaciones. 5.2 Programación funcional Programación funcional en un lenguaje imperativo. Recursión. Lenguajes funcionales. Aplicaciones. 5.3 Programación lógica Fundamentos y origen. Prolog: sintaxis y semántica principal. Aplicaciones. 5.4 Programación paralela Introducción al procesamiento paralelo. Lenguajes de programación que soporten la computación paralela: C, C++, Fortran, Java. Introducción a GPUs y CUDA. Aplicaciones

ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE

Estrategias de aprendizaje sugeridas (Marque con X)		
Aprendizaje basado en problemas	☒	Nemotecnia

Estrategias de aprendizaje sugeridas (Marque con X)			
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 parciales y/o prácticas en el laboratorio de cómputo	25%
- Trabajos y tareas fuera del aula	25%
- Exposición de seminarios por los alumnos	25%
- Participación en clase	25%
Total	100%

PERFIL DEL PROFESOR

Doctor con experiencia en el área computación.

REFERENCIAS BIBLIOGRÁFICAS

Básicas:

- Ross, S. M. 1994. A first course in probability. Ed. Macmillan College Publishing Company. Estados Unidos.
- Rincón, L. 2007. Curso intermedio de probabilidad. Ed. UNAM, Facultad de Ciencias. México.
- Ash, R. B. 2012. Basic probability theory. Dover Publications.
- Verma, S. P. (2005). Estadística Básica para el Manejo de Datos Experimentales. Universidad Nacional Autónoma de México, 185 p.
- Verma, S. P. (2015). Estadística Avanzada para el Manejo de Datos Experimentales. Universidad Nacional Autónoma de México, en prensa, 368 p.

Complementarias:

- Loudon, K. 2011. Programming languages: principles and practice. 3a edición. Ed. Cengage Learning.
- Sebesta, R. 1998. Concepts of programming language. Ed. Addison-Wesley.
- Horowitz, E. 1984. Fundamentals of programming languages. Ed. Computer Science Press.
- Sethi, S. 1989. Programming languages: concepts and constructs. Ed. Addison Wesley.

Web:-----

Otros: -----