

IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE

Unidad académica: Instituto de Investigación en Ciencias Básicas y Aplicadas							
Plan de estudios: Licenciatura en Inteligencia Artificial							
Unidad de aprendizaje: INTRODUCCIÓN A LA PROGRAMACIÓN				Ciclo de formación: Básico Eje general de formación: Teórico-Técnica Área de conocimiento: Fundamentos teóricos de la computación Semestre: 1 ^o			
Elaborada por: Dr. Bruno Lara Guzmán				Fecha de elaboración: Abril, 2021			
Clave:	Horas teóricas :	Horas prácticas :	Horas totales :	Créditos :	Tipo de unidad de aprendizaje :	Carácter de la unidad de aprendizaje :	Modalidad:
IP04CB03000 6	03	00	03	06	Obligatoria	Teórica	Escolarizada
Plan (es) de estudio en los que se imparte: A partir de todos los programas impartidos por el Instituto de Investigación en Ciencias Básicas y Aplicadas.							

ESTRUCTURA DE LA UNIDAD DE APRENDIZAJE

Presentación: En esta unidad de aprendizaje se le da a conocer al estudiantado un panorama general de las Ciencias de la Computación. Asimismo, se sientan las bases y se aportan las herramientas necesarias para traducir problemas a un lenguaje abstracto usando los dos métodos más comunes: diagramas de flujo y pseudo-código.
Propósito: Conozca y distinga los elementos básicos de la computación, su historia, evolución y actualidad, así como los componentes de un sistema de cómputo, los sistemas operativos y los lenguajes de programación, al finalizar la unidad de aprendizaje, mediante el uso de un lenguaje de



programación, para que sea capaz de abstraer partes de un problema y poder solucionarlo, con responsabilidad social.

Competencias que contribuyen al perfil de egreso

Competencias genéricas:

- CG15. Capacidad para formular y gestionar proyectos.
- CG18. Capacidad para actuar en nuevas situaciones.
- CG20. Capacidad de expresión y comunicación.

Competencias específicas:

- CE4. Analiza soluciones computacionales mediante la aplicación de fundamentos teóricos del diseño de algoritmos y estructuras de datos adecuadas para resolver problemas con pensamiento crítico.
- CE5. Descubre nuevas tecnologías, herramientas y estándares en su área de especialidad, mediante la consulta continua de las actualizaciones para mantenerse a la vanguardia en su campo con autonomía y creatividad.

CONTENIDOS

Bloques:	Temas:
1. Introducción	1.1. Elementos físicos de una computadora 1.1.1. Unidad de procesamiento Central 1.1.2. Memoria principal 1.1.3. Dispositivos de entrada/salida 1.2. El Software 1.2.1. Software del sistema 1.2.2. Software de aplicación 1.2.3. Sistema Operativo 1.3. Historia de la computación



	<p>1.4. Arquitectura de una computadora: von Neuman y no Von Neuman</p> <p>1.5. Historia de los Sistemas operativos</p> <p>1.6. Lenguajes, intérpretes y compiladores</p> <p>1.7. Generaciones de los Lenguajes de programación</p>
2. Sistemas de numeración	<p>2.1. Sistemas de numeración</p> <p>2.1.1. Decimal</p> <p>2.1.2. Binario</p> <p>2.1.3. Octal</p> <p>2.1.4. Hexadecimal</p> <p>2.2. Conversión entre diferentes sistemas de numeración</p> <p>2.3. Operaciones básicas en sistemas no decimales</p> <p>2.3.1. Suma</p> <p>2.3.2. Resta</p> <p>2.3.3. Multiplicación</p>
3. Resolución de problemas	<p>3.1. Soluciones analíticas y algorítmicas</p> <p>3.2. Metodología para la solución de problemas por computadora</p> <p>3.2.1. Definición del problema</p> <p>3.2.2. Análisis del problema</p> <p>3.2.3. Diseño del algoritmo</p> <p>3.2.4. Codificación</p> <p>3.2.5. Verificación y depuración</p> <p>3.2.6. Documentación y mantenimiento</p> <p>3.3. Propiedades de un algoritmo</p> <p>3.4. Componentes principales de un algoritmo: entrada, salida, identificación de variables, decisiones y ciclos</p> <p>3.5. Formas de expresar un algoritmo</p> <p>3.5.1. Diagramas de flujo</p> <p>3.5.2. Pseudo-código</p>



	3.5.3. Lenguaje de programación 3.6. Lógica en soluciones algorítmicas
4. Aproximaciones para el diseño de algoritmos	4.1 Modularidad 4.1.1 Aproximación top-down 4.1.2 Aproximación bottom-up 4.2 Recursividad 4.2.1 Recursividad directa e indirecta 4.2.2. Recursión versus Iteración 4.2.3. Resolución de problemas complejos con recursividad
5. Solución de problemas usando un lenguaje de programación	5.1 Tipos de datos. 5.2 Manejo de variables. 5.3 Operadores. 5.4 Estructuras de control; codificación de las estructuras de control en el lenguaje. 5.5 Funciones o subrutinas.

ESTRATEGIAS DE ENSEÑANZA - APRENDIZAJE

Estrategias de aprendizaje sugeridas (Marque X)			
Aprendizaje basado en problemas	(X)	Nemotecnia	()
Estudios de caso	(X)	Análisis de textos	()
Trabajo colaborativo	(X)	Seminarios	()
Plenaria	()	Debate	()
Ensayo	()	Taller	()
Mapas conceptuales	()	Ponencia científica	()
Diseño de proyectos	(X)	Elaboración de síntesis	()
Mapa mental	()	Monografía	()
Práctica reflexiva	()	Reporte de lectura	()
Trípticos	()	Exposición oral	()
Otros			



Estrategias de enseñanza sugeridas (Marque X)			
Presentación oral (conferencia o exposición) por parte del docente	(X)	Experimentación (prácticas)	()
Debate o Panel	()	Trabajos de investigación documental	()
Lectura comentada	()	Anteproyectos de investigación	()
Seminario de investigación	()	Discusión guiada	()
Estudio de Casos	(X)	Organizadores gráficos (Diagramas, etc.)	()
Foro	()	Actividad focal	()
Demostraciones	()	Analogías	()
Ejercicios prácticos (series de problemas)	(X)	Método de proyectos	()
Interacción la realidad (a través de videos, fotografías, dibujos y software especialmente diseñado).	()	Actividades generadoras de información previa	()
Organizadores previos	()	Exploración de la web	(X)
Archivo	()	Portafolio de evidencias	()
Ambiente virtual (foros, chat, correos, ligas a otros sitios web, otros)	()	Enunciado de objetivo o intenciones	()

CRITERIOS DE EVALUACIÓN

Criterios	Porcentaje
<ul style="list-style-type: none"> • Exámenes parciales • Realización de proyecto • Participación en clase • Tareas 	30% 30% 20% 20%
Total	100 %

PERFIL DEL PROFESORADO



Licenciatura, Maestría o Doctorado en ciencias computacionales, matemáticas o ingeniería en áreas afines a las ciencias computacionales, con experiencia docente en el área.

REFERENCIAS

Básicas:

- Beecher, K. (2017). *Computational Thinking: A beginner's guide to problem-solving and programming*. Swindon: BCS Learning & Development Limited.
- Gleen, J. (2012). *Introducción a la Computación*. Pearson.

