

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: PARADIGMAS Y LENGUAJES DE PROGRAMACIÓN				Ciclo de formación: Básico Eje general de formación: Teórico-Técnica Área de conocimiento: Fundamentos teóricos de la computación Semestre: 1º			
Elaborada por: Dra. Lorena Díaz González				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:
PL05CB040109	04	01	05	09	Obligatoria	Teórico - Práctica	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: Durante la unidad de aprendizaje se estudian los principios de lenguajes de programación y los fundamentos de los paradigmas de programación estructural, funcional, lógica y orientada a objetos, mediante la resolución de problemas en distintos lenguajes tales C (estructural); C++ y Java (Objetos); Haskell y Scala (funcional); Prolog (lógico); Python, Ruby y Julia (multiparadigmas).
Propósito:



Conozca, identifique y aplique los paradigmas y lenguajes de programación, como herramienta en el desarrollo de proyectos de software, para solucionar problemas en el entorno real, con creatividad y pensamiento crítico.

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.
- CE6 Contribuye al desarrollo científico y tecnológico, mediante la aplicación de conocimientos y habilidades computacionales adquiridos para la solución innovadora de problemas planteados en el sector público, privado o social con una actitud ética y responsable.

CONTENIDOS

Bloques	Temas
1. Conceptos generales e historia.	1.1 Panorama histórico. 1.2 Criterios para el diseño de lenguajes.
2. Sintaxis.	2.1 Estructura léxica de los lenguajes de programación 2.2 Gramáticas libres de contexto y BNF 2.3 Árboles de análisis y árboles de sintaxis abstracta 2.4 Ambigüedad, asociatividad y precedencia 2.5 EBNF y diagramas de sintaxis.



3. Semántica.	<p>3.1 Atributos y ligaduras.</p> <p>3.2 Declaraciones, bloques y alcance.</p> <p>3.3 Tabla de símbolos.</p> <p>3.4 Resolución de nombres y sobrecarga.</p> <p>3.5 Asignación, vida útil y medio ambiente.</p> <p>3.6 Variables y constantes.</p> <p>3.7 Alias, referencias pendientes y basura.</p>
4. Tipos de datos.	<p>4.1 Tipos de datos primitivos y compuestos. Tipos de datos definidos por el usuario. Apuntadores y manejo de memoria dinámica.</p> <p>4.2 Equivalencia de tipos.</p>
5. Tipos de datos abstractos y módulos.	<p>5.1 Especificación algebraica de tipos de datos abstractos (TDA).</p> <p>5.2 Mecanismos de TDA en Java y C++.</p>
6. Programación orientada a objetos.	<p>6.1 Conceptos fundamentales, clases, métodos, objetos.</p> <p>6.2 Lenguajes orientados a objetos.</p>
7. Programación funcional.	<p>7.1 Historia y origen del paradigma funcional.</p> <p>7.2 Conceptos fundamentales: recursión, funciones de orden superior y recursión de cola.</p> <p>7.3 Casos de estudios: comparación entre recursividad e iteración.</p> <p>7.4 Lenguajes funcionales.</p> <p>7.5 Matemáticas de la programación funcional: cálculo lambda.</p>
8. Programación lógica.	<p>8.1 Lógica y programas lógicos. Cálculo de predicados de primer orden</p> <p>8.2 Clausulas de Horn.</p> <p>8.3 Resolución y unificación.</p> <p>8.4 Prolog: sintaxis y semántica principal.</p>

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	()	Seminarios	(X)
Plenaria	()	Debate	()
Ensayo	()	Taller	()
Mapas conceptuales	()	Ponencia científica	()
Diseño de proyectos	(X)	Elaboración de síntesis	()
Mapa mental	()	Monografía	()
Práctica reflexiva	()	Reporte de lectura	(X)
Trípticos	()	Exposición oral	()
Otros			
Estrategias de enseñanza sugeridas (Marque X)			
Presentación oral (conferencia o exposición) por parte del docente	(X)	Experimentación (prácticas)	(X)
Debate o Panel	(X)	Trabajos de investigación documental	()
Lectura comentada	(X)	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)	()	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	()
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
• Seminarios	30%
• Proyecto final	30%
• Resolución de problemas y estudios de caso	20%
• Reporte de lecturas	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:

- Jones, C. B. (2020). *Understanding Programming Languages*. Springer International Publishing AG.
- Loudon, K. C., & Lambert, K. A. (2016). *Programming languages: principles and practices*. Cengage Learning.
- Lee, K. D. (2017). *Foundations of programming languages*. Springer.
- Toal, R., Rivera, R., Schneider, A., & Choe, E. (2016). *Programming language explorations*. CRC Press.

Complementarias:

- Samuel, M. S. (2017). An Insight into Programming Paradigms and Their Programming Languages. *Journal of Applied Technology and Innovation*, 1(1), 37-57.
- Priestley, M. (2017). AI and the origins of the functional programming language style. *Minds and Machines*, 27(3), 449-472.
- Kiselyov, O., & King, A. (Eds.). (2016). *Functional and Logic Programming: 13th International Symposium, FLOPS 2016, Kochi, Japan, March 4-6, 2016, Proceedings* (Vol. 9613). Springer.

