

## IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE

<b>Unidad académica:</b> Instituto de Investigación en Ciencias Básicas y Aplicadas							
<b>Plan de estudios:</b> Licenciatura en Inteligencia Artificial							
<b>Unidad de aprendizaje:</b>  PROGRAMACIÓN DE AMBIENTES GRÁFICOS EN C# .NET.				<b>Ciclo de formación:</b> Profesional <b>Eje general de formación:</b> Para la Generación y Aplicación del Conocimiento <b>Semestre:</b> 6º			
<b>Elaborada por:</b> M.I.C.A. Yainier Labrada Nueva				<b>Fecha de elaboración:</b> Abril, 2021			
<b>Clave:</b>	<b>Horas teóricas:</b>	<b>Horas prácticas:</b>	<b>Horas totales:</b>	<b>Créditos:</b>	<b>Tipo de unidad de aprendizaje:</b>	<b>Carácter de la unidad de aprendizaje:</b>	<b>Modalidad:</b>
	03	02	05	08	Optativa	Teórico - Práctica	Escolarizada
<b>Plan (es) de estudio en los que se imparte:</b> A partir de todos los programas impartidos por el Instituto de Investigación en Ciencias Básicas y Aplicadas							

## ESTRUCTURA DE LA UNIDAD DE APRENDIZAJE

<b>Presentación:</b>  Durante la unidad de aprendizaje se estudian principios y fundamentos en el lenguaje de programación C# para interfaces visuales, mediante diversas técnicas basadas en la programación Orientada a Objetos y la Programación Orientada a Eventos.
<b>Propósito:</b>  Conozca, identifique y aplique los conocimientos sobre los principios fundamentales del lenguaje de programación C# y sus características, al término de la unidad de aprendizaje, a través de múltiples aplicaciones a la programación visual escritorio y web, para la resolución de problemas de la vida real, con responsabilidad social y ética profesional.
<b>Competencias que contribuyen al perfil de egreso</b>
<b>Competencias genéricas:</b>  <ul style="list-style-type: none"> <li>CG1. Capacidad para el aprendizaje de forma autónoma.</li> </ul>



<ul style="list-style-type: none"> <li>• CG5. Capacidad de aprender y actualizarse permanentemente.</li> <li>• CG6. Capacidad para la investigación.</li> </ul>
<p><b>Competencias específicas:</b></p> <ul style="list-style-type: none"> <li>• CE11. Desarrolla sistemas computacionales inteligentes utilizando una computadora con la arquitectura y lenguaje de programación adecuados para la resolución de problemas con una actitud investigativa y socialmente responsable.</li> <li>• CE12. Implementa, prueba y mantiene proyectos de sistemas inteligentes empleando criterios de cumplimiento según estándares de calidad establecidos y aprovechando al máximo sus recursos, para resolver problemas científicos y tecnológicos y tomar decisiones que generen bienestar para la sociedad en su conjunto.</li> <li>• CE13. Analiza impactos locales y globales de la Inteligencia Artificial mediante el uso de criterios objetivos utilizando lenguaje técnico apropiado, comunicando efectivamente conceptos, métodos y resultados en forma oral y escrita, para presentar propuestas y proyectos de una manera ética y responsable.</li> </ul>

## CONTENIDOS

Bloques	Temas
1. Introducción.	1.1. La historia de C# 1.2. El marco de trabajo .NET de Microsoft 1.3. ¿Qué es un programa? 1.4. Fundamentos de programación 1.5. Errores comunes de programación 1.6. Ejercicios
2. Entorno de desarrollo de C#.	2.1. Introducción 2.2. Instalación y configuración 2.3. Cree su primer programa 2.4. Los controles en tiempo de diseño 2.5. Los eventos y el control Button 2.6. Apertura de un proyecto existente 2.7. Documentación de los valores de las propiedades 2.8. Errores en los programas 2.9. Funciones del editor



	<ul style="list-style-type: none"> <li>2.10. El cuadro de mensajes</li> <li>2.11. Ayuda</li> <li>2.12. Fundamentos de programación</li> <li>2.13. Errores comunes de programación</li> <li>2.14. Secretos de codificación</li> <li>2.15. Nuevos elementos del lenguaje</li> <li>2.16. Nuevas características del IDE</li> <li>2.17. Ejercicios</li> </ul>
<p>3. Introducción a los gráficos</p>	<ul style="list-style-type: none"> <li>3.1. Introducción</li> <li>3.2. Objetos, métodos, propiedades, clases: una analogía</li> <li>3.3. Herencia <ul style="list-style-type: none"> <li>3.3.1. Uso de la herencia</li> <li>3.3.2. Palabrada reservada <b>protected</b></li> <li>3.3.3. Redefinición</li> <li>3.3.4. Diagramas de clases</li> <li>3.3.5. La herencia base. Constructores. Clases Abstractas.</li> <li>3.3.6. Nuevos elementos del lenguaje C#</li> </ul> </li> <li>3.4. Polimorfismo <ul style="list-style-type: none"> <li>3.4.1. El polimorfismo en acción</li> <li>3.4.2. Conceptos asociados al polimorfismo</li> </ul> </li> <li>3.5. Nuestro primer dibujo</li> <li>3.6. Creación del programa</li> <li>3.7. El sistema de coordenadas de gráficos</li> <li>3.8. Explicación del programa</li> <li>3.9. Métodos para dibujar</li> <li>3.10. Colores</li> <li>3.11. El concepto de secuencia y las instrucciones</li> <li>3.12. Adición de significado mediante el uso de comentarios</li> <li>3.13. Fundamentos de programación</li> <li>3.14. Errores comunes de programación</li> <li>3.15. Secretos de codificación</li> </ul>



	3.16. Nuevos elementos del lenguaje 3.17. Nuevas características del IDE 3.18. Ejercicios
4. Manejo de Excepciones	4.1. ¿Qué es una Excepción? 4.2. Manejando Excepciones utilizando el bloque Try/Catch 4.3. Utilizando un bloque Finally 4.4. Lanzamiento de Excepciones

### ESTRATEGIAS DE ENSEÑANZA - APRENDIZAJE

Estrategias de aprendizaje sugeridas (Marque X)			
Aprendizaje basado en problemas	( X )	Nemotecnia	( )
Estudios de caso	( )	Análisis de textos	( )
Trabajo colaborativo	( X )	Seminarios	( )
Plenaria	( )	Debate	( X )
Ensayo	( )	Taller	( )
Mapas conceptuales	( )	Ponencia científica	( )
Diseño de proyectos	( X )	Elaboración de síntesis	( )
Mapa mental	( )	Monografía	( )
Práctica reflexiva	( )	Reporte de lectura	( )
Trípticos	( )	Exposición oral	( X )
Otros			
Estrategias de enseñanza sugeridas (Marque X)			
Presentación oral (conferencia o exposición) por parte del docente	( X )	Experimentación (prácticas)	( X )
Debate o Panel	( )	Trabajos de investigación documental	( )
Lectura comentada	( X )	Anteproyectos de investigación	( )



Seminario de investigación	( )	Discusión guiada	( X )
Estudio de Casos	( )	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	( )
Archivo	( )	Portafolio de evidencias	( )
Ambiente virtual (foros, chat, correos, ligas a otros sitios web, otros)	( )	Enunciado de objetivo o intenciones	( X )

### **CRITERIOS DE EVALUACIÓN**

<b>Criterios</b>	<b>Porcentaje</b>
<ul style="list-style-type: none"> <li>• Exámenes parciales</li> <li>• Realización de práctica</li> <li>• Búsqueda de información</li> <li>• Participación en clase</li> <li>• Reportes</li> </ul>	<p>30%</p> <p>30%</p> <p>20%</p> <p>10%</p> <p>10%</p>
<b>Total</b>	100 %

### **PERFIL DEL PROFESORADO**

Licenciatura, Maestría o Doctorado en el área afín a la disciplina de la unidad de aprendizaje (Ciencias Computacionales)



## REFERENCIAS

### Básicas:

- J. Pitt-Francis and J. Whiteley. (2017). *Guide to Scientific Computing in C++*.
- M. Muñoz-Serafín. (2017). *Introducción al C# Manual del Estudiante*.
- J. ferguson, B. Patterson, J. Beres, P. Boutquin and M. Gupta. (2003). *La Biblia de C#*.
- D. Bell and M. Parr. (2010). *C# Para Estudiantes*.
- N.A. Landa-Cosio. (2010). *C# La Guía Total del Programador*.

### Complementarias:

- Downey, A. B., & Mayfield, C. (2019). *Think Java: How to Think Like a Computer Scientist*. O'Reilly Media.
- Hunt, J. (2018). *A Beginner's Guide to Scala, Object Orientation and Functional Programming*. Springer.

