

### 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 ORIENTADA A OBJETOS Y EVENTOS				<b>Ciclo de formación:</b> Básico <b>Eje general de formación:</b> Teórico-Técnica <b>Área de conocimiento:</b> Fundamentos teóricos de la computación <b>Semestre:</b> 3º			
<b>Elaborada por:</b> Dra. Lorena Díaz González				<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>
PO19CB03000 6	03	00	03	06	Obligatoria	Teórica	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 los fundamentos del paradigma de programación orientado a objetos, mediante la resolución de casos de estudio en el lenguaje Java.
<b>Propósito:</b>  Comprenda y aplique los fundamentos del paradigma de programación orientado a objetos usando el lenguaje de programación Java, como herramienta en el desarrollo de proyectos de software, para innovar y desarrollar soluciones computacionales con responsabilidad y compromiso.



<b>Competencias que contribuyen al perfil de egreso</b>
<p><b>Competencias genéricas:</b></p> <ul style="list-style-type: none"> <li>• CG15. Capacidad para formular y gestionar proyectos.</li> <li>• CG18. Capacidad para actuar en nuevas situaciones.</li> <li>• CG20. Capacidad de expresión y comunicación.</li> </ul>
<p><b>Competencias específicas:</b></p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> </ul>

## **CONTENIDOS**

<b>Bloques</b>	<b>Temas</b>
1. Introducción a las clases, objetos, métodos y cadenas.	1.1 Tipos de lenguajes de programación orientados a objetos (POO): C++, java, ADA y <i>Python</i> . 1.2 Declaración de clases; creación de objetos de una clase; declaración de métodos. 1.3 Variables de instancia; métodos set (establecer) y get (obtener). 1.4 Inicialización de objetos mediante constructores. 1.5 Comparación entre tipos primitivos y tipos por referencia.
2. Métodos: un análisis detallado.	2.1 Métodos y campos estáticos. 2.2 Métodos con múltiples parámetros; declarar y utilizar los métodos; pila de llamadas a los métodos y los registros de activación. 2.3 Promoción y conversión de argumentos; alcance de las declaraciones y sobrecarga de métodos.
3. Clases y objetos: un análisis más detallado	3.1 Control del acceso a los miembros. 3.2 Constructores sobrecargados, constructores predeterminados y sin argumentos. 3.3 Composición, enumeraciones y recolección de basura.



	3.4 Miembros de clase static, declaración static import y variables de instancia final.
4. Herencia, polimorfismo e interfaces.	4.1 Herencia: superclases y subclases; miembros protegidos (Java); relaciones entre superclases y subclases; constructores en las subclases. 4.2 Polimorfismo: clases y métodos abstractos. 4.3. Interfaces: Creación y manejo de interfaces; Interfaces comunes de la API de Java.
5. Manejo de excepciones.	5.1 Generalidades acerca del manejo de excepciones en Java y jerarquía de excepciones. 5.2 Manejo de excepciones.
6. Interfaces gráficas de usuario	6.1 Generalidades del manejo de evento. 6.2 Construcción de interfaces gráficas de usuario GUIs. 6.3 Administradores de esquemas. 6.4 Uso de paneles y marcos como administradores complejos. 6.5 Manejo de eventos de acción, ratón y teclado. 6.6 Interfaces de escucha. Clases adaptadoras. Manejo de eventos usando clases internas anónimas.
7. Introducción a las clases, objetos, métodos y cadenas.	7.1 Estados, prioridades y programación de procesos. 7.2 Creación y ejecución de procesos: objetos <i>Runnable</i> y la clase <i>Thread</i> . 7.3 Sincronización de procesos: compartir datos con y sin sincronización. 7.4 Procesamiento múltiple con GUIs: <i>SwingWorker</i> .
8. Métodos: un análisis detallado.	8.1 Tecnologías de programación funcional. 8.2. Interfaces funcionales. 8.3. Expresiones Lambda. 8.4. Streams.

### 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	( )	Elaboración de síntesis	( )



Mapa mental	( )	Monografía	( )
Práctica reflexiva	( )	Reporte de lectura	( X )
Trípticos	( )	Exposición oral	( )
Otros			
<b>Estrategias de enseñanza sugeridas (Marque X)</b>			
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)	( X )	Enunciado de objetivo o intenciones	( )

### CRITERIOS DE EVALUACIÓN

Crterios	Porcentaje
• Seminarios	30%
• Proyecto final	30%
• Resolución de problemas y estudios de caso	30%
• Reporte de lecturas	10%
<b>Total</b>	<b>100 %</b>

### 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:**

- Deitel, P. J., & Deitel, H. M. (2017). *Java: How to Program, Late Objects* (p. 1248). New York City, NY: Pearson.
- Ogiwara, M. (2018). *Fundamentals of Java Programming*. Springer.
- Sznajdleder, P. (2018). *Java a fondo: estudio del lenguaje y desarrollo de aplicaciones*. Alfaomega Grupo Editor.

### **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.

