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: LABORATORIO DE PROGRAMACIÓN ORIENTADA A OBJETOS Y EVENTOS				**Ciclo de formación:** Básico **Eje general de formación:** En Contexto **Semestre:** 3º			
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:**
LP24CB000202	00	02	02	02	Obligatoria	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 aplican los fundamentos del paradigma de programación orientado a objetos, mediante la resolución de casos de estudio en el lenguaje Java.
Propósito: 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.
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. • CE9. Aplica conocimientos, habilidades y actitudes adquiridas mediante la realización de proyectos multidisciplinarios para promover aprendizajes significativos de manera constructiva y participativa.

CONTENIDOS

Bloques	**Temas**
1. Introducción a las clases, objetos, métodos y cadenas.	1.1 Declaración de clases; creación de objetos de una clase; declaración de métodos; variables de instancia; métodos set (establecer) y get (obtener); inicialización de objetos mediante constructores; comparación entre tipos primitivos y tipos por referencia. 1.2 Casos de estudio y resolución de ejercicios.
2. Métodos: un análisis detallado.	2.1 Métodos y campos estáticos; métodos con múltiples parámetros; declarar y utilizar los métodos; pila de llamadas a los métodos y los registros de activación; promoción y conversión de argumentos; alcance de las declaraciones y sobrecarga de métodos. 2.2 Caso de estudio y resolución de ejercicios.
3. Clases y objetos: un análisis más detallado	3.1 Control del acceso a los miembros; constructores sobrecargados, constructores predeterminados y sin argumentos; composición, enumeraciones y recolección de basura; miembros de clase *static*, declaración *static import* y variables de instancia *final*. 3.2 Caso de estudio y resolución de ejercicios.
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; creación y manejo de interfaces. 4.3 Caso de estudio y resolución de ejercicios.
5. Manejo de excepciones.	5.1 Manejo de excepciones en Java.

	5.2 Caso de estudio y resolución de ejercicios.
6. Interfaces gráficas de usuario	6.1 Construcción de interfaces gráficas de usuario GUIs; aplicación de administradores de esquemas; uso de paneles y marcos como administradores complejos; manejo de eventos de acción, ratón y teclado; interfaces de escucha; clases adaptadoras; manejo de eventos usando clases internas anónimas. 6.2 Caso de estudio y resolución de ejercicios.
7. Concurrencia.	7.1 Estados, prioridades y programación de procesos; creación y ejecución de procesos: objetos *Runnable* y la clase *Thread; s*incronización de procesos: compartir datos con y sin sincronización; procesamiento múltiple con GUIs: *SwingWorker*. 7.2 Caso de estudio y resolución de ejercicios.
8. Lambdas y streams de Java.	8.1 Tecnologías de programación funcional; interfaces funcionales; expresiones Lambda; *streams*. 8.2. Caso de estudio y resolución de ejercicios.

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	( )
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	( )	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)	( )	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
• Resolución de problemas y casos de estudio • Proyecto final	50% 50%
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:

- Deitel, P. J., & Deitel, H. M. (2017). *Java: How to Program, Late Objects* (p. 1248). New York City, NY: Pearson.
- Ogihara, 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.