

**DESCRIPCIÓN DE LA ASIGNATURA ALGORITMOS Y PROGRAMACIÓN**

<b>Facultad:</b>	Ingeniería	<b>Departamento:</b>	Gestión de Proyectos y Sistemas
<b>Código:</b>	BPTSP05	<b>Asignatura:</b>	Algoritmos y Programación
<b>Créditos:</b>	3	<b>Tipo:</b>	X Obligatoria ___ Electiva
<b>Carreras:</b>	Ingeniería de Sistemas (IS) Ingeniería de Producción (IP) Ingeniería Química (IQ) Ingeniería Mecánica (IM) Ingeniería Eléctrica (IE) Ingeniería Civil (IC) Matemáticas Industriales (MI)	<b>Trimestres:</b>	III y IV
<b>Prerrequisito</b>	Pensamiento Computacional (FBTSP04)	<b>Modalidad:</b>	Presencial
<b>Número de horas semanales</b>			
<b>En aula</b>	<b>Prácticas supervisadas</b>	<b>Laboratorio</b>	<b>Aprendizaje Autónomo</b>
4		1	4
<b>Coordinador:</b>	Christian Guillén Drija	<b>Fecha de actualización</b>	Dic 2024

- Justificación:** Esta asignatura es indispensable para la formación de todo ingeniero en la UNIMET, ya que proporciona las herramientas para desarrollar un pensamiento lógico-computacional aplicable a diversas disciplinas. A través del aprendizaje de la lógica de programación, los estudiantes de Ingeniería de Sistemas, Producción, Química, Mecánica, Eléctrica y Civil adquieren la capacidad de utilizar el computador como un recurso clave en la resolución de problemas.
- Propósito:** Proporcionar a los estudiantes de ingeniería de la UNIMET las herramientas y habilidades fundamentales para desarrollar soluciones computacionales efectivas. A través del estudio de la lógica de programación y la implementación de algoritmos en un lenguaje de programación de alto nivel, los estudiantes aprenderán a abordar y resolver problemas de ingeniería complejos, estableciendo una base sólida para su desarrollo profesional en cualquier campo de la ingeniería.
- Objetivos:**  
Diseñar soluciones algorítmicas eficientes y estructuradas para problemas de ingeniería, aplicando principios de pensamiento lógico y algorítmico, y utilizando herramientas de representación como diagramas de flujo o pseudocódigo. Relacionado con los objetivos de la carrera 1 y 2.  
Implementar soluciones algorítmicas en un lenguaje de programación de alto nivel, demostrando un dominio de las estructuras de control, tipos de datos y técnicas de programación orientada a objetos, incluyendo el concepto de encapsulamiento. Relacionado con los objetivos de la carrera 1 y 2.  
Analizar problemas de ingeniería de diversa complejidad, identificando los requisitos, restricciones y posibles soluciones, y aplicando técnicas de análisis de problemas para descomponerlos en subproblemas más manejables. Relacionado con los objetivos de la carrera 1 y 2.
- Resultados de aprendizaje:**  
**RA5: Manejo de las TICs.** Capacidad para utilizar herramientas y recursos tecnológicos actualizados para potenciar la construcción de conocimiento pertinente en un contexto cultural amplio. (N2)  
**RA6: Innovación y Creatividad.** Capacidad para combinar los conocimientos básicos y los especializados propios del área de estudio para generar propuestas innovadoras y competitivas en la actividad profesional. (N2)
- Contenido:**

Tema	Contenido	Herramientas técnicas y actividades	Horas dedicadas
1	Bases conceptuales de la Programación Orientada a Objetos: Introducción a Programación Orientada a Objetos. Concepto de Encapsulamiento. Clases, objetos y mensajes	ABP: Modelado UML e implementación en lenguaje de alto nivel (Ej. Sistema de Biblioteca). Debate comparativo: POO vs. Programación Estructurada. Práctica deliberada de diseño.	6
2	Programación Orientada a Objetos: · Métodos de un objeto. Constructores y selectores.	Estudio de caso: Diseño e implementación de sistema de inventarios. Aprendizaje colaborativo (Parejas). Desafíos de codificación intensiva y tutorías dirigidas.	20

	<ul style="list-style-type: none"> <li>· Tipos de datos, variables, expresiones y asignaciones</li> <li>· Estructuras de decisión</li> <li>· Estructuras de repetición</li> <li>· Entrada de datos</li> <li>· Pase de parámetros</li> </ul>		
3	Recursividad	ABP: Resolución de problemas de ingeniería (Árboles/Fractales). Laboratorio de experimentación de patrones recursivos. Proyecto de visualización de algoritmos.	10
4	Ordenamiento y Búsqueda: <ul style="list-style-type: none"> <li>· Métodos de ordenamiento y búsqueda.</li> <li>· Manejo de Arreglos.</li> </ul>	Implementación de algoritmos y análisis de eficiencia temporal/espacial. Uso de simuladores. Proyecto: Aplicación de búsqueda/ordenamiento en problemas de ingeniería.	12

6. **Métodos de aprendizaje:** Aprendizaje basado en problemas (ABP): Presentar a los estudiantes problemas de ingeniería del mundo real que requieran la aplicación de algoritmos y programación para su solución. Aprendizaje colaborativo: Fomentar el trabajo en equipo a través de proyectos, debates y actividades en grupo que permitan a los estudiantes compartir ideas, resolver dudas y aprender de forma colaborativa. Tutorías: Ofrecer sesiones de tutoría individual o en grupos pequeños para atender dudas específicas y brindar apoyo personalizado a los estudiantes. Aprendizaje basado en proyectos: Desarrollar proyectos de programación que permitan a los estudiantes aplicar los conocimientos adquiridos en la creación de soluciones informáticas. Uso de plataformas de aprendizaje en línea: Utilizar plataformas como Moodle, y Replit para complementar las clases presenciales. Consideraciones generales: Se recomienda el uso de herramientas de control de versiones como Git para el manejo del código fuente de los proyectos. Se pueden utilizar herramientas de visualización de algoritmos para facilitar la comprensión de los conceptos.

#### 7. Métodos de evaluación:

Aprendizaje en contacto con el docente ( 40%)	Aprendizaje práctico experimental (40%)	Aprendizaje autónomo (20 %)
Exposiciones, Participación en clases, Debates, Exámenes escritos u orales, Talleres, Defensa de proyectos, entre otros.	Resolución de problemas prácticos, Prácticas de laboratorio, Salidas de campo o visitas técnicas, Manejo de software especializado, Prototipado técnico, Estudios de caso técnicos, entre otros.	Elaboración de informes, Resolución de problemas y ejercicios, Ensayos de investigación, Creación de mapas conceptuales, Participación en foros, entre otros.

#### 8. Referencias

##### Obligatoria:

- Deitel, H., & Deitel, P. (2004). *Java: cómo programar*. Prentice Hall.
- Joyanes Aguilar, L. (2003). *Fundamentos de programación I, algoritmos, estructura de datos y objetos*. McGraw-Hill.
- Joyanes Aguilar, L., Castillo Sanz, A., Sánchez García, L., & Zahonero Martínez, I. (2005). *Algoritmos, programación y estructura de datos* (1a ed.). McGraw-Hill.
- Joyanes Aguilar, L., & Zahonero, I. (2002). *Programación en Java 2, algoritmos, estructura de datos y programación orientada a objetos*. McGraw-Hill.
- Waite, M., & Lafore, R. (2005a). *Data structures and algorithms*. Waite Group Press.
- Waite, M., & Lafore, R. (2005b). *Object-oriented design in Java*. Waite Group Press.