



PLAN DE ESTUDIOS (PE): Licenciatura en Matemáticas

ÁREA: Interdisciplinaria

ASIGNATURA: Algoritmos, Estructuras de datos y Objetos

CÓDIGO:

CRÉDITOS: 6

FECHA: Julio de 2017





1. DATOS GENERALES

Nivel Educativo:	Licenciatura
Nombre del Plan de Estudios:	Licenciatura en Matemáticas
Modalidad Académica:	Presencial
Nombre de la Asignatura:	Algoritmos, Estructuras de Datos y Objetos
Ubicación:	Nivel formativo
Correlación:	
Asignaturas Precedentes:	Computación, Programación I, Programación II
Asignaturas Consecuentes:	Temas Selectos de Computación

2. CARGA HORARIA DEL ESTUDIANTE

Concepto	Horas por semana		Total de horas por periodo	Total de créditos por periodo
	Teoría	Práctica		
Horas teoría y práctica (16 horas = 1 crédito)	2	3	100	6

3. REVISIONES Y ACTUALIZACIONES

Autores:	Edgar Santiago Moyotl Hernández, Mónica Macías Pérez, Patricia Domínguez Soto
Fecha de diseño:	Julio de 2017
Fecha de la última actualización:	No aplica
Fecha de aprobación por parte de la academia de área, departamento u otro.	6 de julio del 2017





Revisores:	Edgar Santiago Moyotl Hernández, Mónica Macías Pérez, Patricia Domínguez Soto, Sergio Adán Juárez
Sinopsis de la revisión y/o actualización:	Asignatura de nueva creación, la cual completa la formación básica en el área de Cómputo Científico.

4. PERFIL DESEABLE DEL PROFESOR (A) PARA IMPARTIR LA ASIGNATURA:

Disciplina profesional:	Matemáticas, Matemáticas Aplicadas y Ciencias de la computación
Nivel académico:	Maestría
Experiencia docente:	1 año
Experiencia profesional:	1 año

5. PROPÓSITO: El alumno reconocerá y aplicará las técnicas de diseño de algoritmos y los conceptos de estructura de datos para resolver problemas reales por medio de la computadora, además, reconocerá los conceptos fundamentales de la Programación Orientada a Objetos.

6. COMPETENCIAS PROFESIONALES:

- Reconocer y evaluar las principales estructuras de datos para manipular los datos de un algoritmo apropiadamente.
- Evaluar la eficiencia y complejidad de un algoritmo, así como seleccionar el que mejor se adapte a las necesidades del problema a resolver.
- Distinguir los diferentes tipos de algoritmos de ordenación y contrastar sus resultados.
- Describir los conceptos fundamentales de la Programación Orientada a Objetos y aplicar dichos conceptos en el desarrollo de programas.
- Desarrollar la habilidad de implementar diferentes tipos de algoritmos y seleccionar en forma apropiada el algoritmo más eficiente para un problema dado.





7. CONTENIDOS TEMÁTICOS

Unidad de Aprendizaje	Contenido Temático	Referencias
1. Estructuras de datos	<ol style="list-style-type: none"> 1. Tipo de Dato Abstracto (TDA) 2. Estructuras lineales <ol style="list-style-type: none"> 2.1. Listas ligadas 2.2. Listas circulares 2.3. Listas doblemente ligadas 2.4. Pilas 2.5. Colas 3. Estructuras no lineales <ol style="list-style-type: none"> 3.1. Árboles binarios 3.2. Árboles balanceados 3.3. Grafos 3.4. Montículos (Heaps) 4. Tablas de dispersión (Hash) 	<p>Baase, S. y Van, G. A. (2002). <i>Algoritmos computacionales. Introducción al Análisis y Diseño</i> (3ra. ed.). México: Pearson Educación.</p> <p>Ceballos, J. (2015). <i>C/C++ Curso de Programación</i> (4ta. ed.). México: Alfaomega</p>
2. Técnicas de ordenamiento	<ol style="list-style-type: none"> 1. Complejidad algorítmica 2. Ordenación por inserción 3. Ordenación por selección 4. Ordenación por mezcla (Mergesort) 5. Ordenación mediante montículos (Heapsort) 6. Ordenación rápida (Quicksort) 	<p>Deitel, P. J. y Deitel, H. M. (2010). <i>C how to program</i> (6th. ed.). México: Prentice Hall.</p> <p>Joyanes, Aguilar, L. (2014). <i>Programación en C, C++, Java y UML</i> (2da. ed.). México: McGraw-Hill.</p> <p>Joyanes, Aguilar L. (2008). <i>Fundamentos de Programación</i> (4ta. ed.). España: McGraw Hill.</p>
3. Técnicas de diseño de algoritmos	<ol style="list-style-type: none"> 1. Divide y vencerás 2. Algoritmos voraces 3. Programación dinámica 4. Vuelta atrás 5. Ramificación y poda 	<p>López, J. L. y Gutiérrez, A. (2014). <i>Programación orientada a objetos con C++ y Java</i>. México: Grupo Editorial Patria.</p>
4. Programación Orientada a Objetos	<ol style="list-style-type: none"> 1. Objetos 2. Clases 3. Abstracción 4. Encapsulamiento y ocultación de datos 5. Herencia 6. Reutilización o reusabilidad 7. Polimorfismo 	<p>Murillo, M., Gómez, L. y Castillo, F. (2006). <i>Introducción a la programación orientada a objetos</i>. México: Pearson Educación.</p>





8. ESTRATEGIAS, TÉCNICAS Y RECURSOS DIDÁCTICOS

Estrategias y técnicas didácticas	Recursos didácticos
<ul style="list-style-type: none"> ● Lluvia de ideas ● Aprendizaje basado en problemas ● Aprendizaje orientado a proyectos ● Aprendizaje cooperativo ● Aprendizaje colaborativo ● Prácticas ● Elaboración de programas ● Investigación bibliográfica extraclase 	<p>Materiales:</p> <ul style="list-style-type: none"> ● Impreso: libros y fotocopias. ● Digital: libros, artículos y diapositivas. ● Pizarrón, plumones y borrador. ● Proyector y computadora. ● Software libre: Octave, Python, Herramientas para programar en C y/o C++. ● Páginas web, correo electrónico, chats y foros.

9. EJES TRANSVERSALES

Eje (s) transversales	Contribución de la asignatura
Formación Humana y Social	Solucionar problemas reales promueve la participación del alumno de manera cooperativa y colaborativa.
Desarrollo de Habilidades en el uso de las Tecnologías de la Información y la Comunicación	El uso de herramientas computacionales promueve el uso de medios electrónicos.
Desarrollo de Habilidades del Pensamiento Complejo	Mediante la programación se desarrolla la habilidad de resolver problemas conceptuales y cuantitativos utilizando diferentes formas de razonamiento (lógico, aritmético, algebraico y analógico).
Lengua Extranjera	La bibliografía y el uso de los lenguajes de programación en idioma inglés desarrollan la habilidad lectora y de comprensión de textos escritos en otro idioma.
Innovación y Talento Universitario	Resolver problemas científicos ayuda a que el alumno desarrolle la habilidad para crear soluciones innovadoras y generar cambios.
Educación para la Investigación	Mediante la revisión del análisis y diseño de algoritmos se desarrollan las habilidades necesarias para el ejercicio de la investigación en el cómputo científico.





10. CRITERIOS DE EVALUACIÓN

Criterios	Porcentaje
▪ Exámenes	50%
▪ Participación en clase	10%
▪ Tareas	10%
▪ Prácticas	20%
▪ Proyecto final	10%
Total	100%

11. REQUISITOS DE ACREDITACIÓN

Estar inscrito como alumno en la Unidad Académica en la BUAP
Asistir como mínimo al 80% de las sesiones para tener derecho a exentar por evaluación continua y/o presentar el examen final en ordinario o extraordinario
Asistir como mínimo al 70% de las sesiones para tener derecho al examen extraordinario
Cumplir con las actividades académicas y cargas de estudio asignadas que señale el PE

