



PLAN DE ESTUDIOS (PE): Licenciatura en Matemáticas

ÁREA: Interdisciplinaria

ASIGNATURA: Programación I

CÓDIGO: MATS 011

CRÉDITOS: 6

FECHA: Noviembre 2016





1. DATOS GENERALES

Nivel Educativo:	Licenciatura
Nombre del Plan de Estudios:	Licenciatura en Matemáticas
Modalidad Académica:	Presencial
Nombre de la Asignatura:	Programación I
Ubicación:	Nivel básico
Correlación:	
Asignaturas Precedentes:	Computación
Asignaturas Consecuentes:	Ninguna

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 Sergio Adán Juárez Patricia Domínguez Soto
Fecha de diseño:	Diciembre 2009
Fecha de la última actualización:	Noviembre 2015
Fecha de aprobación por parte de la academia de área, departamento u otro.	Diciembre 2016
Revisores:	Edgar Santiago Moyotl Hernández Mónica Macías Pérez Sergio Adán Juárez Aureliano Jorge Jiménez Martínez Elizabeth Martínez Banfi Mauricio Esteban Chacón Tirado
Sinopsis de la revisión y/o actualización:	Se describieron las competencias profesionales a desarrollar, se reorganizaron las unidades de aprendizaje, se actualizó el contenido y la bibliografía.

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

Disciplina profesional:	Ciencias de la computación, Ingeniería en sistemas computacionales, Matemáticas o carreras afines.
Nivel académico:	Maestría
Experiencia docente:	1 año
Experiencia profesional:	1 año

5. PROPÓSITO: El alumno desarrollará habilidades para analizar problemas computacionales, diseñar soluciones a través de algoritmos e implementar programas con el lenguaje de programación C, aplicando las técnicas de la programación estructurada.





6. COMPETENCIAS PROFESIONALES:

- Identificar las etapas para la resolución de un problema.
- Definir formalmente lo que es un algoritmo, un lenguaje de programación, un programa y describir el proceso de compilación y ejecución.
- Identificar las propiedades y estructura de un algoritmo para representarlo en pseudocódigo y/o diagramas de flujo.
- Definir los conceptos de las estructuras de control: secuencial, selectiva y repetitiva para aplicarlos al resolver problemas con toma de decisiones y/o naturaleza iterativa.
- Aplicar los conceptos elementales de variable, tipos de datos, operadores, entrada/salida, asignación, estructuras de control del lenguaje C para escribir programas que den solución a distintos problemas bajo el enfoque de la programación estructurada.
- Definir el diseño descendente y aplicar el concepto de función para escribir programas que se pueden modularizar.
- Definir el concepto de estructura de datos y aplicar el concepto de arreglo en problemas que requieren almacenar conjuntos de datos del mismo tipo.
- Evaluar la importancia y el impacto de la programación en el contexto de las matemáticas.

7. CONTENIDOS TEMÁTICOS

Unidad de Aprendizaje	Contenido Temático	Referencias
1. Introducción a la programación	1. Programación de computadoras 2. Definición de algoritmo y programa 3. Lenguajes de programación 4. Fases de la programación 4.1. Análisis de un problema 4.2. Diseño de un algoritmo 4.3. Implementación de un programa	Cairó Battistutti, O. G. (2006). <i>Fundamentos de programación. Piensa en C</i> (1ra. ed.). México: Pearson Educación.
2. Algoritmia	1. Algoritmos 1.1. Propiedades 1.2. Estructura 1.3. Representación 2. Conceptos fundamentales	Ceballos, J. (2015). <i>C/C++ Curso de Programación</i> (4ta. ed.). México: Alfaomega.





Unidad de Aprendizaje	Contenido Temático	Referencias
	<ul style="list-style-type: none"> 2.1. Tipos de datos 2.2. Literales 2.3. Variables 2.4. Constantes 2.5. Identificadores 2.6. Entrada y salida de datos 2.7. Expresiones <ul style="list-style-type: none"> 2.7.1. Operador de asignación 2.7.2. Operadores aritméticos 2.7.3. Prioridad y asociatividad de operadores 	<p>Corona, M. A. y Ancona, M. A. (2011). <i>Diseño de algoritmos y su codificación en lenguaje C</i>. México: McGraw-Hill.</p> <p>Deitel, P. J. y Deitel, H. M. (2010). <i>C how to program</i> (6th. ed.). México: Prentice Hall.</p>
3. Estructuras de programación	<ul style="list-style-type: none"> 1. Estructura secuencial 2. Estructura de selección <ul style="list-style-type: none"> 2.1. Condiciones: operadores relacionales y lógicos 2.2. Simple 2.3. Doble 2.4. Anidada 2.5. Múltiple 3. Estructura de repetición <ul style="list-style-type: none"> 3.1. Evaluación previa 3.2. Evaluación posterior 3.3. Anidada 4. Tipos de ciclos <ul style="list-style-type: none"> 4.1. Controlados por contador 4.2. Controlados por suceso 	<p>Joyanes Aguilar, L. (2008). <i>Fundamentos de programación</i> (4ta. ed.). España: McGraw-Hill.</p> <p>Joyanes Aguilar, L. (2014). <i>Programación en C, C++, Java y UML</i> (2da. ed.). México: McGraw-Hill.</p>
4. Lenguaje de programación C (Parte I)	<ul style="list-style-type: none"> 1. Estructura de un programa en C 2. Implementación de un programa en C <ul style="list-style-type: none"> 2.1. Entornos de desarrollo 2.2. Edición 2.3. Compilación 2.4. Ejecución 2.5. Depuración y tipos de errores 3. Elementos principales <ul style="list-style-type: none"> 3.1. Archivos cabecera 3.2. Función principal 3.3. Comentarios 3.4. Tipos de datos y modificadores de 	





Unidad de Aprendizaje	Contenido Temático	Referencias
	<p>tipos</p> <p>3.5. Identificadores y palabras reservadas</p> <p>3.6. Declaración de variables</p> <p>3.7. Literales y constantes</p> <p>3.8. Entrada y salida de datos</p> <p>3.9. Expresiones</p> <p>3.10. Operador de asignación</p> <p>3.11. Operadores aritméticos</p> <p>3.12. Operadores de incremento y decremento</p> <p>3.13. Operadores relacionales</p> <p>3.14. Operadores lógicos</p> <p>3.15. Prioridad y asociatividad de operadores</p>	
<p>5. Lenguaje de programación C (Parte II)</p>	<p>1. Instrucciones</p> <p>1.1. Simples</p> <p>1.2. Compuestas o bloques</p> <p>2. Instrucciones de control</p> <p>2.1. Secuencial</p> <p>2.2. Selectiva</p> <p>2.2.1. <i>if</i></p> <p>2.2.2. <i>if-else</i></p> <p>2.2.3. <i>if</i> anidado</p> <p>2.2.4. <i>switch</i></p> <p>2.2.5. Operador ternario</p> <p>2.3. Repetitiva</p> <p>2.3.1. <i>while</i></p> <p>2.3.2. <i>do-while</i></p> <p>2.3.3. <i>for</i></p> <p>2.3.4. ciclos anidados</p> <p>2.3.5. Instrucciones <i>break</i> y <i>continue</i></p> <p>3. Subprogramas</p> <p>3.1. Funciones</p> <p>3.2. Funciones intrínsecas y extrínsecas</p> <p>3.3. Ámbito: variables locales y globales</p> <p>3.4. Paso de parámetros</p> <p>3.4.1. Por valor</p> <p>3.4.2. Por referencia</p> <p>4. Estructuras de datos</p> <p>4.1. Números pseudo-aleatorios</p>	





Unidad de Aprendizaje	Contenido Temático	Referencias
	4.2. Arreglos unidimensionales 4.3. Arreglos bidimensionales 4.4. Cadenas 4.5. Algoritmos de ordenación y búsqueda 4.6. Medición experimental de la eficiencia de los algoritmos de ordenamiento	

8. ESTRATEGIAS, TÉCNICAS Y RECURSOS DIDÁCTICOS

Estrategias y técnicas didácticas	Recursos didácticos
<ul style="list-style-type: none"> • Resúmenes • Paráfrasis • Mapas conceptuales • Lluvia de ideas • Aprendizaje basado en problemas • Aprendizaje orientado a proyectos • Aprendizaje cooperativo • Aprendizaje colaborativo • Responder a preguntas exploratorias y literales • Prácticas • Elaboración de programas • Investigación bibliográfica extraclase • Portafolio electrónico 	Materiales: <ul style="list-style-type: none"> • Impreso: libros y fotocopias. • Digital: libros, artículos y diapositivas. • Pizarrón, plumones y borrador. • Proyector y computadora. • Programas informáticos: se sugiere Raptor, JavaBlock, PSeInt, DevC++, Code::Blocks y/o Zinjai. • 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 software para programar promueve el uso de medios electrónicos.
Desarrollo de Habilidades del Pensamiento Complejo	Mediante la algoritmia se desarrolla la habilidad de resolver problemas conceptuales y cuantitativos utilizando diferentes formas de razonamiento (lógico, aritmético, algebraico y analógico).





Lengua Extranjera	Dado que la gran mayoría de los lenguajes de programación están en idioma inglés, se desarrolla la habilidad lectora y de comprensión de textos escritos en otro idioma.
Innovación y Talento Universitario	Resolver problemas computacionales ayuda a que el alumno desarrolle la habilidad para crear soluciones innovadoras y generar cambios.
Educación para la Investigación	Mediante la programación estructurada se orienta a una cultura de indagación, descubrimiento y construcción de conocimientos nuevos.

10. CRITERIOS DE EVALUACIÓN

Criterios	Porcentaje
▪ Exámenes	50%
▪ Participación en clase	10%
▪ Tareas	10%
▪ Prácticas de laboratorio	10%
▪ Proyecto final	20%
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

