PLAN DE ESTUDIOS (PE): Licenciatura en Matemáticas Aplicadas

ÁREA: Análisis y Métodos Numéricos

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 Aplicadas
Modalidad Académica:	Presencial
Nombre de la Asignatura:	Programación I
Ubicación:	Nivel básico
Correlación:	
Asignaturas Precedentes:	Computación
Asignaturas Consecuentes:	Programación II, Análisis y Métodos Numéricos I, Análisis y Métodos Numéricos II

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 seudocódigo 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). *Fundamentos de programación. Piensa en C* (1ra. ed.). México: Pearson Educación. Ceballos, J. (2015). C/C++ *Curso de Programación* (4ta. ed.). México: Alfaomega.
2. Algoritmia	1. Algoritmos 1.1. Propiedades 1.2. Estructura 1.3. Representación 2. Conceptos fundamentales	

Unidad de Aprendizaje	Contenido Temático	Referencias
	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 2.7.1. Operador de asignación 2.7.2. Operadores aritméticos 2.7.3. Prioridad y asociatividad de operadores	Corona, M. A. y Ancona, M. A. (2011). *Diseño de algoritmos y su codificación en lenguaje C*. México: McGraw-Hill. Deitel, P. J. y Deitel, H. M. (2010). *C how to program* (6th. ed.). México: Prentice Hall.
3. Estructuras de programación	1. Estructura secuencial 2. Estructura de selección 2.1. Condiciones: operadores relacionales y lógicos 2.2. Simple 2.3. Doble 2.4. Anidada 2.5. Múltiple 3. Estructura de repetición 3.1. Evaluación previa 3.2. Evaluación posterior 3.3. Anidada 4. Tipos de ciclos 4.1. Controlados por contador 4.2. Controlados por suceso	Joyanes Aguilar, L. (2008). *Fundamentos de programación* (4ta. ed.). España: McGraw-Hill. Joyanes Aguilar, L. (2014). *Programación en C, C++, Java y UML* (2da. ed.). México: McGraw-Hill.
4. Lenguaje de programación C (Parte I)	1. Estructura de un programa en C 2. Implementación de un programa en C 2.1. Entornos de desarrollo 2.2. Edición 2.3. Compilación 2.4. Ejecución 2.5. Depuración y tipos de errores 3. Elementos principales 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
	tipos 3.5. Identificadores y palabras reservadas 3.6. Declaración de variables 3.7. Literales y constantes 3.8. Entrada y salida de datos 3.9. Expresiones 3.10. Operador de asignación 3.11. Operadores aritméticos 3.12. Operadores de incremento y decremento 3.13. Operadores relacionales 3.14. Operadores lógicos 3.15. Prioridad y asociatividad de operadores	
5. Lenguaje de programación C (Parte II)	1. Instrucciones 1.1. Simples 1.2. Compuestas o bloques 2. Instrucciones de control 2.1. Secuencial 2.2. Selectiva 2.2.1. *if* 2.2.2. *if-else* 2.2.3. *if* anidado 2.2.4. *switch* 2.2.5. Operador ternario 2.3. Repetitiva 2.3.1. *while* 2.3.2. *do-while* 2.3.3. *for* 2.3.4. ciclos anidados 2.3.5. Instrucciones *break* y *continue* 3. Subprogramas 3.1. Funciones 3.2. Funciones intrínsecas y extrínsecas 3.3. Ámbito: variables locales y globales 3.4. Paso de parámetros 3.4.1. Por valor 3.4.2. Por referencia 4. Estructuras de datos 4.1. Números seudo-aleatorios	

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
• 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: • 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%delas sesiones para tener derecho al examen extraordinario
Cumplir con las actividades académicas y cargas de estudio asignadas que señale el PE