



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Escuela Nacional de Estudios Superiores Unidad Juriquilla



Plan de Estudios de la
Licenciatura en Órtesis y Prótesis

| | | | | | | | | |
|------------------------------|---|----------------------|-----------------------------|------------------|--------------|------------------|--------------|----------------|
| Programa Programación | | | | | | | | |
| Clave | Semestre 4 | Créditos 6 | Duración | | 16 semanas | | | |
| | | | Área de conocimiento | | Tecnología | | | |
| | | | Etapa de formación | | Intermedia | | | |
| Modalidad | Curso (X) Taller () Lab () Sem () | | | | Tipo | T () | P () | T/P (X) |
| Carácter | Obligatorio (X) Optativo () | | | | Horas | | | |
| | Obligatorio E () Optativo E () | | | | | | | |
| | | | | Semana | | Semestre | | |
| | | | | Teóricas | 2 | Teóricas | 32 | |
| | | | | Prácticas | 2 | Prácticas | 32 | |
| | | | | Total | 4 | Total | 64 | |

| | |
|-------------------------------|--|
| Seriación | |
| Ninguna (X) | |
| Obligatoria () | |
| Asignatura antecedente | |
| Asignatura consecuente | |
| Indicativa () | |
| Asignatura antecedente | |
| Asignatura consecuente | |

Objetivo general:

El alumno resolverá problemas aplicando los fundamentos de programación para diseñar programas en el lenguaje estructurado C, apoyándose en metodologías para la solución de problemas.

Objetivos específicos:

- 1 Definir la importancia de la programación como herramienta.
- 2 Resolver problemas mediante la especificación algorítmica
- 3 Construir programas utilizando el lenguaje de programación C a través de un análisis y modelado algorítmico previo.
- 4 Distinguir los diversos paradigmas de programación;
- 5 Identificar la aplicación del cómputo para la solución de problemas

Índice temático

| | | | |
|---|------------------|-----------------------|------------------|
| | Tema | Horas Semestre | |
| | | Teóricas | Prácticas |
| 1 | Panorama general | 2 | 0 |

| | | | |
|---|---|----|----|
| 2 | Resolución de problemas | 10 | 10 |
| 3 | Fundamentos para la construcción de código a partir del algoritmo | 10 | 12 |
| 4 | Paradigmas de programación | 5 | 5 |
| 5 | Cómputo aplicado a diferentes disciplinas | 5 | 5 |
| | | 32 | 32 |
| | Subtotal | | |
| | Total | | 64 |

Contenido Temático

| Tema | Subtemas |
|------|---|
| 1 | <p>Panorama general</p> <p>1.1 Evolución de la programación.</p> <p>1.2 Beneficios de la programación (a la sociedad, a la industria, a la medicina, entre otros).</p> <p>1.3 Algoritmos en la solución de problemas y sus retos.</p> <p>1.4 Explicar el propósito y el papel de los fundamentos de la programación en la ingeniería.</p> |
| 2 | <p>Resolución de problemas</p> <p>2.1 Definición, planteamiento y modelado del problema.</p> <p style="padding-left: 20px;">2.1.1 Formular el problema.</p> <p style="padding-left: 20px;">2.1.2 Analizar el problema.</p> <p style="padding-left: 20px;">2.1.3 Diseñar una estrategia de búsqueda de la solución.</p> <p>2.2 Algoritmos para la resolución del problema</p> <p style="padding-left: 20px;">2.2.1 Definición y representación de algoritmos.</p> <p style="padding-left: 20px;">2.2.2 Conversión del planteamiento del problema al algoritmo.</p> <p>2.3 Definición del modelo computacional.</p> <p style="padding-left: 20px;">2.3.1 Máquina de Von Neuman.</p> <p style="padding-left: 20px;">2.3.2 Máquina de Turing.</p> <p>2.4 Refinamiento de algoritmo paso a paso</p> <p style="padding-left: 20px;">2.4.1 Planteamiento de la solución del problema</p> <p style="padding-left: 20px;">2.4.2 Descomposición de la solución del problema en submódulos.</p> <p style="padding-left: 20px;">2.4.3 Aplicación de las estructuras básicas de control: secuencial, condicional e iterativo.</p> |
| 3 | <p>Fundamentos para la construcción de código a partir del algoritmo</p> <p>3.1 Sintaxis básica y semántica</p> <p>3.2 Variables, tipos, expresiones y asignación.</p> <p>3.3 Estructuras de control condicional e iterativo.</p> <p>3.4 Funciones y paso de parámetros.</p> <p>3.5 Descomposición estructurada.</p> <p>3.6 Manejo de E/S</p> <p>3.7 Estrategias de depuración.</p> <p style="padding-left: 20px;">3.7.1 Tipo de errores</p> <p style="padding-left: 20px;">3.7.2 Técnicas de depuración</p> |
| 4 | <p>Paradigmas de programación</p> <p>4.1 Programación estructurada.</p> <p>4.2 Programación orientada a objetos</p> <p>4.3 Programación lógica</p> <p>4.4 Programación paralela.</p> <p>4.5 Principales usos de los paradigmas para la solución de problemas</p> <p>4.6 Nuevas tendencias</p> |
| 5 | <p>Cómputo aplicado a diferentes disciplinas</p> <p>5.1 Tendencia de desarrollo de software</p> <p style="padding-left: 20px;">5.1.1 Software propietario.</p> <p style="padding-left: 20px;">5.1.2 Software libre.</p> |

| 5.2 Aplicaciones | | | |
|--|--|--|-----|
| 5.2.1 | Ciencias físicas y de la ingeniería. | | |
| 5.2.2 | Ciencias médicas y de la salud. | | |
| 5.2.3 | Leyes, ciencias sociales y del comportamiento. | | |
| 5.2.4 | Artes y humanidades. | | |
| 5.2.5 | Otras disciplinas | | |
| | | | |
| | | | |
| Estrategias didácticas | | Evaluación del aprendizaje | |
| Exposición | (X) | Exámenes parciales | (X) |
| Trabajo en equipo | (X) | Examen final | (X) |
| Lecturas | (X) | Trabajos y tareas | (X) |
| Trabajo de investigación | (X) | Presentación de tema | () |
| Prácticas (taller o laboratorio) | () | Participación en clases | (X) |
| Prácticas de campo | () | Asistencia | (X) |
| Aprendizaje por proyectos | () | Rúbricas | () |
| Aprendizaje basado en problemas | (X) | Portafolios | () |
| Casos de enseñanza | (X) | Listas de cotejo | () |
| Otras (especificar) | () | Otras (especificar) | () |
| Perfil profesiográfico. | | | |
| Título o Grado | | Licenciatura en Ingeniería, Informática, Ciencias de la Computación o carreras cuyo contenido en el área sea similar. Deseable haber realizado estudios de posgrado. | |
| Experiencia docente | | Debe contar con experiencia docente o haber participado en cursos o seminarios de iniciación en la práctica docente. | |
| Otra característica | | El profesor que impartirá la asignatura deberá ser, preferentemente, académico de la UNAM con área de competencia y trabajo afín a la disciplina. La asignatura puede ser impartida por un profesor o investigador de tiempo completo o de asignatura con actividad profesional o académica directamente relacionada con el programa de la asignatura y con su aplicación profesional. | |
| Bibliografía básica | | | |
| Brookshear, J. G. (2011). <i>Computer Science: An Overview</i> . 11a. ed. Boston: Prentice Hall. | | | |
| Cairó, O. (2003). <i>Metodología de la programación. Algoritmos, diagramas de flujo y programas. Tomos I y II</i> . 2a. ed. México: Alfaomega | | | |
| Felleisen, M. Findlet, R.B. et al. (2001). <i>How to design programs. An introduction to programming and computing</i> . Cambridge: MIT Press. | | | |
| Horowitz, E. (2007). <i>Computer Algorithms</i> . 2a. ed. Summit, NJ: Silicon Press. | | | |
| Kerhghan, B.W. & Pike, R. (1994). <i>The Practice of programming</i> . New Jersey: Addison-Wesley. | | | |
| Kerhghan, B.W. & Ritchie, D. (1988). <i>C Programming Language</i> . 2a. ed. New Jersey: Prentice Hall. | | | |
| McConnell, S. (2004). <i>Code Complete 2</i> . Redmond, WA: Microsoft Press | | | |
| Sznajdleder, P. (2012). <i>Algoritmos a fondo: con implementación en C y JAVA</i> . Buenos Aires: Alfaomega. | | | |
| Volland, G. (2003). <i>Engineering by design</i> . 2a. ed. Upper Saddle River, NJ: Prentice Hall. | | | |

Bibliografía complementaria

Allen, T. & Noonan, R. (2006). *Programming languages*. 2a. ed. New Jersey: McGraw-Hill.

Michalel, L.S. (2009). *Programming language pragmatics*. 3a. ed. Cambridge: Morgan Kaufmann.

Peter. S. *Programming language concepts (Undergraduate topics in computer science)*. Copenhagen: Springer.