



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
LICENCIATURA EN CIENCIAS GENÓMICAS

PROGRAMA DE ASIGNATURA

| CLAVE | NOMBRE DE LA ASIGNATURA | | | SEMESTRE | |
|-----------|----------------------------|----------------|--------------------------------------|----------|----------|
| | PRINCIPIOS DE PROGRAMACIÓN | | | PRIMERO | |
| MODALIDAD | CARÁCTER | HORAS SEMESTRE | HORA/SEMANA TEÓRICAS PRÁCTICAS | | CRÉDITOS |
| CURSO | OBLIGATORIA | 80 | 3 | 2 | 8 |
| NIVEL | BÁSICO | | | | |
| TIPO | TEÓRICO-PRÁCTICA | | | | |

OBJETIVO: Que el alumno conozca las bases y antecedentes de la computación, así como que comprenda los conceptos básicos de programación, adquiriendo un pensamiento abstracto que le permita resolver problemas mediante la implementación de programas computacionales. Que el alumno se inicie en la programación mediante la implementación de programas computacionales en diversos lenguajes de programación.

| Número de Horas: | Contenidos temáticos: |
|------------------|---|
| 10 | 1. Introducción a la computación 1.1 ¿Qué es una computadora? 1.2 ¿Qué es un algoritmo? 1.3 ¿Qué es un lenguaje de programación? 1.4 ¿Qué tipos de lenguajes de programación existen? 1.5 ¿Qué es un paradigma de programación y cuales existen? 1.6 ¿Cuál es la metodología para solucionar problemas con una computadora? 1.7 Lenguajes algorítmicos. |
| 15 | 2. Programación estructurada 2.1 Variables. 2.2 Operadores. 2.3 Estructuras secuenciales. 2.4 Estructuras de control de flujo. |
| 20 | 3. Módulos (subrutinas, funciones, procedimientos) 3.1 Ámbito de una variable (variables globales y locales). 3.2 Diferencia entre procedimientos y funciones. 3.3 Declaración de procedimientos y funciones. 3.4 Paso de argumentos por valor y por referencia. 3.5 Recursividad. |
| 15 | 4. Estructuras de datos básicas 4.1 Arreglos lineales (vectores). 4.2 Matrices y arreglos multidimensionales. 4.3 Pilas y Colas. 4.4 Árboles y Grafos (redes). |
| 10 | 5. Introducción a lenguajes y máquinas 5.1 ¿Que relación guardan los lenguajes con la computabilidad? 5.2 Jerarquía de Chomsky de los lenguajes. 5.3 Autómatas finitos. 5.4 Gramáticas transformacionales. 5.5 Autómatas pushdown y gramáticas libres de contexto (CFG). |
| 10 | 6. Implementación de programas computacionales en Perl y C |
| 80 | <i>Total de Horas</i> |

BIBLIOGRAFÍA BÁSICA:

- Wirth, N. *Algoritmos y Estructuras de Datos*. Prentice Hall, 1987.

- Lipschutz, S. *Estructura de Datos*. Serie Schaum en Computación. McGraw-Hill, 1988.
- Wirth, N. *Algoritmos + Estructuras de Datos = Programas*. Dossat, 1992.
- Dijkstra, E.W. *A Discipline of Programming*. Prentice Hall, 1997.
- Knuth, D.E. *The Art of Computer Programming Vol. 1, 2, and 3* (3rd Ed.). Addison Wesley, 1997.
- Sudkamp, T. A. *Languages and Machines: An Introduction to the Theory of Computer Science* (2nd Ed.). Addison-Wesley, 1998.
- Hopcroft, J. E., Motwani, R. and Ullman, J. D. *Introduction to Automata Theory, Languages and Computation* (2nd Ed.). Addison-Wesley Longman, 2000.
- Kernighan, B.W., y Ritchie, D.M. *El Lenguaje de Programación C* (2da Ed.). Prentice Hall, 1991.
- Schwartz, R.L., and Phoenix, T. *Learning Perl* (3rd Ed.). O'Reilly, 2001.
- Wall, L., Christiansen, T., and Orwant, J. *Programming Perl* (3rd Ed.). O'Reilly, 2000.
- Ralston, A., Reilly, E. D. and Hemmendinger, D. *Encyclopedia of Computer Science* (4th Ed.). Nature Publishing Group, 2000.

BIBLIOGRAFÍA COMPLEMENTARIA:

- Turing, A. (1950). Computing machinery and intelligence. *Mind* LIX: 433-460.
- Benenson Y., Adar R., Paz-Elizur T., Livneh Z., Shapiro E. DNA molecule provides a computing machine with both data and fuel. *Proc. Natl. Acad. Sci. U.S.A.* **100(5)**, 2191-2196. 2003.
- Benenson Y, Gil B, Ben-Dor U, Adar R, Shapiro E. An autonomous molecular computer for logical control of gene expression. *Nature*. **429(6990)**, 423-429. 2004.
- Hayes, B. Computing comes to life. *American Scientist*. **89(3)**, 204-208. 2001.
- Chen J., Wood D.H. Computation with biomolecules. *Proc. Natl. Acad. Sci. U.S.A.* **97(4)**, 1328-1330. 2000.
- Hayes, B. Graph Theory in Practice: Part I. *American Scientist*. **88(1)**, 9-13. 2000.
- Hayes, B. Graph Theory in Practice: Part II. *American Scientist*. **88(2)**, 104-109. 2000.
- Ravasz E., Somera A.L., Mongru D.A., Oltvai Z.N., Barabasi A.L. Hierarchical organization of modularity in metabolic networks. *Science*. **297(5586)**, 1551-1555. 2002.
- Agar, J. and Turney, J. *Turing and the Universal Machine: The Making of the Modern Computer*. Totem Books Press, 2001.
- Aho, A. V. and Hopcroft, J. E. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Longman, 1974.
- Atallah, M. J., Fox, S. and Lassandro, S. *Algorithms and Theory of Computation Handbook*. CRC Press, 1998.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. *Introduction to Algorithms* (2nd Ed.). The MIT Press, 2001.
- Baase, S. and Van Gelder, A. *Computer Algorithms: Introduction to Design and Analysis* (3rd Ed.). Addison-Wesley Longman, 1999.
- Kingston, J. H. *Algorithms and Data Structures: Design, Correctness* (2nd Ed.). Addison-Wesley Longman, 1997.
- Knuth, D. E. and Greene, D. H. *Mathematics for the Analysis of Algorithms* (3rd Ed.). Birkhauser Boston Press, 1999.
- Okasaki, C. *Purely Functional Data Structures*. Cambridge University Press, 1999.
- Skiena, S. S. *The Algorithm Design Manual*. Springer-Verlag, 1997

SUGERENCIAS DIDÁCTICAS:

Exposición de los temas por parte del profesor, con la participación activa de los estudiantes.
Ejercicios en clase y de tarea.

SUGERENCIAS DE EVALUACIÓN:

Exámenes teóricos.

Ejercicios.

Participación en clase.

Proyecto final.

PERFIL PROFESIOGRÁFICO:

Licenciado(a) en Ciencias Genómicas; Maestro(a) o Doctor(a) en Ciencias de la Computación o Ingeniería en Sistemas Computacionales.