



Facultad de Ciencias Naturales, Exactas y de la  
Educación

Departamento: Matemáticas

Tipo de Actividad: Asignatura

Créditos: 5 por semestre

Nombre: Complejidad Computacional I ( Mat 561 )

Intensidad Horaria: 4 h.s.

Requisitos: Mat 364

Co-requisitos:

#### DESCRIPCIÓN DEL CURSO

Inicialmente se presentan los elementos básicos de computación, y algunos modelos de computación, tales como las máquinas de Turing determinísticas y No determinísticas y las máquinas de acceso aleatorio. La *Teoría de la Complejidad Computacional* clasifica un problema de acuerdo al tipo de modelo computacional, con recursos acotados, necesario para resolverlo. Así, se tienen las siguientes clases: P (respectivamente NP) como la clase de problemas resolubles por máquinas de Turing determinísticas (respectivamente no determinísticas) que utilizan tiempo polinomial; o equivalentemente, en términos de algoritmos convencionales, en la clase P están los problemas de decisión que se resuelven en tiempo polinomial y en la clase NP están los problemas de decisión que se verifican en tiempo polinomial. Estas clases cumplen con la relación:  $P \subseteq NP$ . La pregunta de si la contención anterior es propia es uno de los problemas, aun abiertos, más importantes de la Teoría de la Complejidad Computacional, y su resolución otorgará fama y gloria a su autor.

#### OBJETIVOS GENERALES

Desarrollar habilidades para clasificar formalmente los problemas en las diferentes clases de complejidad.

#### OBJETIVOS ESPECÍFICOS

1. Conocer algunos modelos de computación y las respectivas relaciones entre ellos.
2. Clasificar los problemas según la naturaleza inherente de los mismos.
3. Identificar las relaciones entre las diferentes clases de complejidad.
4. Ampliar los conocimientos del estudiante en el análisis y diseño de algoritmos.
5. Conocer las principales aplicaciones de la teoría de grafos.
6. Reconocer el análisis y diseño de algoritmos como una herramienta tecnológica.
7. Identificar las condiciones que deben cumplir los problemas de programación para poder aplicar las diferentes técnicas de programación.
8. Utilizar las diferentes técnicas de programación en la construcción de algoritmos eficientes.

#### METODOLOGÍA

##### Teoría:

Se realizarán exposiciones por parte del profesor, siguiendo básicamente, los libros incluidos en la bibliografía. En algunos temas se asignarán lecturas complementarias y talleres que permitan consolidar y profundizar los conocimientos adquiridos.

##### Práctica:

Se realizarán clases prácticas en el laboratorio de sistemas. Habrá una clase práctica de 2 horas cada semana. En ellas se implementarán ejercicios que complementen y refuercen el aprendizaje de los contenidos impartidos en las clases teóricas.

Los alumnos deben desarrollar los programas de los ejercicios propuestos por el profesor en el laboratorio. Al finalizar el curso, los alumnos deben desarrollar un proyecto donde apliquen los conocimientos adquiridos en la presente asignatura. Se debe entregar copia del proyecto en medio magnético, y realizar la sustentación correspondiente del trabajo.

#### CONTENIDO

##### CAPITULO I. LENGUAJES

- 1.1 Cadenas y lenguajes
- 1.2 Expresiones y conjuntos regulares

##### CAPITULO II: MODELOS BÁSICOS DE COMPUTACIÓN.

- 2.1 Máquinas de Turing determinísticas
  - 2.1.1 Máquina de Turing de una cinta
  - 2.1.2 Máquina de Turing de  $k$ -cintas
- 2.2 Máquinas de Turing No determinísticas.
  - 2.2.1 Modelo estándar
  - 2.2.2 Modelo no estándar

- 2.3 Máquina de Turing Universal
- 2.4 El Problema de la parada
- 2.5 Máquina de acceso aleatorio (RAM)

#### **CAPITULO III: NP-COMPLETITUD**

- 3.1 La Clase P
- 3.2 La Clase NP.
  - 3.2.1 Máquinas de Turing No determinísticas y la Clase NP
  - 3.2.2 Algoritmos de verificación y la Clase NP
- 3.3 Relación Entre P y NP
- 3.4 Transformación tiempo polinomial y la Clase NP.
- 3.5 Teorema de Cook.

#### **CAPITULO IV: ALGUNOS PROBLEMAS NP-COMPLETOS**

- 3.1 Variantes de SAT.
- 3.2 Problemas de teoría de grafos.
- 3.3 Problemas de Recubrimientos.
- 3.4 Técnicas de demostración
  - 3.4.1 Restricción

#### **CAPITULO IV: APROXIMABILIDAD**

- 3.1 Problemas NPO: Definiciones y preliminares.
- 3.2 Problemas de decisión subyacentes.
- 3.3 La clase APX.
- 3.4 La clase PTAS.
- 3.5 La clase FPTAS.
- 3.6 Reducciones y completitud.

#### **CAPITULO V: LÓGICA Y COMPLEJIDAD**

- 4.1 Expresiones booleanas.
- 4.2 Satisfabilidad y validez.
- 4.3 Funciones booleanas y circuitos
- 4.4 La sintaxis de la lógica de primer orden.
- 4.5 Expresiones válidas.
- 4.6 Algunos elementos de la lógica de segundo orden.
- 4.7 Caracterizaciones de la lógica: Teorema de Fagin.

#### **BIBLIOGRAFÍA:**

1. Daniel Pierre Bovet, Pierluigi Crescenzi. *Introduction to the theory complexity*. Prentice Hall, International, 1994.
2. Christos H. Papadimitriou, *Computational complexity*. Addison Wesley, Publishing Company, 1994.
3. Dorit S. Hochbaum. *Approximation algorithms for NP-hard Problem*. PWS Publishing Company, 1997.
4. Michael R. Garey, David S. Johnson. *Computer and Intractability. A Guide to theory of NP-Completeness*. W. H. Freeman and Company, 1979.
5. Thomas Cormen, Charles Leiserson, Ronald Rivest. *Introduction to algorithms*. MIT Press, 1990.
6. G. Brassard, P. Bratley. *Fundamentos de algoritmos*. Prentice Hall, 1997.
7. Hebert S. Wilf. *Algorithms and complexity*. Prentice Hall, 1986.