



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Programa de Complejidad Computacional

1. NOMBRE DE LA UNIDAD CURRICULAR

Complejidad Computacional

2. CRÉDITOS

9 (nueve)

3. OBJETIVOS DE LA UNIDAD CURRICULAR

Que el estudiante conozca que tipo de limitaciones intrínsecas fundamentales hay (en espacio, tiempo, aleatoriedad, etc.) en diversos problemas computacionales. Que el estudiante conozca diversos métodos para estudiar la dificultad intrínseca de diversos problemas computacionales. Dar fundamentos, ejemplos y aplicaciones de la complejidad computacional.

4. METODOLOGÍA DE ENSEÑANZA

El curso tiene una visión integral teórico-práctico. En este sentido, los conceptos dados en clase van a estar ilustrados con una gran cantidad de ejemplos y de ejercicios que se irán realizando tanto en clase como en ejercicios obligatorios.

Los ejercicios obligatorios se realizarán en dos entregas (una a la mitad y otro al fin del curso) y son individuales. Varias clases teórico-prácticas van a estar dedicadas a discutir estos ejercicios obligatorios.

Las clases son interactivas con mucha participación estudiantil y con una dinámica consistente en dejar material de lectura para discutir en la clase siguiente. En las clases se va a hacer énfasis en aspectos fundamentales de cada tema que no están presentes en los libros, pero que van a ayudar sustancialmente a entender los temas cuando los estudiante lean el material. Por tal motivo la asistencia a clase es obligatoria en el entendido de que da mucho valor agregado para el aprovechamiento del curso.

Horas clases teórico-prácticas:	60 hs.
Horas de estudio para clases:	30 hs.
Horas trabajo obligatorio:	45 hs.
Total:	135 hs.

5. TEMARIO

1. Introducción: Modelos computacionales, Máquinas de Turing, clases computacionales.
2. Complejidad en el tiempo. Clases P y NP. Reducción de problemas. Problemas NP-Completos. Teorema de Cook. Reducciones de Karp. Clase CO-NP, EXP y NEXP.
3. Complejidad en el espacio. Clases PSPACE y NPSPACE. Teorema de Savitch. Clases L y NL. Problemas PSPACE completos. Funciones logspace implícitas. Problemas NL COMPLETOS.
4. Jerarquía polinomial. Máquinas de Turing alternantes. Balance de tiempo y espacio para SAT.
5. Computación probabilística. Máquinas de Turing probabilistas. Clases RP, ORP y ZPP. Clase BPP y relación con otras clases. Reducciones probabilísticas.

6. BIBLIOGRAFÍA

Tema	Básica	Complementaria
Introducción: Modelos computacionales, Máquinas de Turing, clases computacionales.	(1)	(2,3)
Complejidad en el tiempo. Clases P y NP. Problemas NP-Completos.	(1)	(2,3)
Complejidad en el espacio. Clases PSPACE y NPSPACE. Teorema de Savitch. Clases L y NL. Problemas PSPACE completos y NL completos.	(1)	(2,3)
Jerarquía polinomial.	(1)	(3)
Computación probabilística	(1)	(3)

6.1 Básica

1. Sanjeev Arora y Boaz Barak (2009). Computational Complexity: A Modern Approach. Cambridge University Press. ISBN-13: 9780521424264. 2009

Hay copias del libro en la Biblioteca del Pedeciba Informática.

6.2 Complementaria

2. Michael Sipser (2013). Introduction to the Theory of Computation (3rd Edition). CENGAGE Learning. ISBN-13: 978-1133187790. 2013.

Este libro está en la Biblioteca del Pedeciba Informática.

3. Oded Goldreich (2008). Computational Complexity: A Conceptual Perspective. Cambridge University Press. ISBN-13: 978-0-521-88473-0. 2008.

Este libro está en la Biblioteca del Pedeciba Informática.

7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS

7.1 Conocimientos Previos Exigidos:

Lenguajes formales y autómatas.

7.2 Conocimientos Previos Recomendados:

Conocimientos básicos de estructuras de datos y algoritmos.

ANEXO A

A1) INSTITUTO

Computación.

A2) CRONOGRAMA TENTATIVO

Semana 1	Introducción. ¿Porqué surge la necesidad de una teoría de la complejidad computacional? Modelos de computación (4 hs.)
Semana 2	Máquinas de Turing, clases computacionales (4 hs.)
Semana 3	Práctico sobre el punto 1 del temario (2 hs.) Complejidad en tiempo (2 hs.)
Semana 4	Clases P y NP (2 hs.) Reducción de Problemas (2 hs.)
Semana 5	Problemas NP-completos, Teorema de Cook, Reducciones de Karp (4 hs.)
Semana 6	Clase CO-NP, EXP y NEXP (2 hs.) Práctico sobre el punto 2 del temario y Obligatorio 1 (2 hs.)
Semana 7	Complejidad en espacio (2 hs.) PSPACE, NPSPACE. Teorema de Savitch (2 hs.)
Semana 8	Problema PSPACE completos (2 hs.) Clase L y NL. Funciones logspace implícitas (2 hs.)
Semana 9	Problemas NLCOMPLETOS (2 hs.) Consultas Obligatorio 1 (2 hs.)
Semana 10	Práctico sobre el punto 3 del temario (2 hs.) Jerarquía Polinomial (2 hs.)
Semana 11	Máquinas de Turing alternantes (2 hs.) Balance de tiempo y espacio para SAT (2 hs.)
Semana 12	Práctico sobre el punto 4 del temario y Obligatorio 2 (2 hs.) Computación probabilística (2 hs.)
Semana 13	Máquina de Turing Probabilistas (2 hs.) Clases RP, CORP y ZPP (2 hs.)
Semana 14	Clase BPP y relación con otras clases (2 hs.) Reducciones probabilistas (2 hs.)
Semana 15	Práctico sobre el punto 5 del temario (2 hs.) Consultas finales del Obligatorio 2 (2 hs.)

A3) MODALIDAD DEL CURSO Y PROCEDIMIENTO DE EVALUACIÓN

Este es un curso exigente y con alta participación estudiantil en las clases. En general se va a mencionar en la clase anterior el material a leer, y en la clase siguiente se realizará una interacción con los estudiantes para responder dudas y relacionar los temas leídos con los ya vistos anteriormente (o en incluso en otros cursos de la carrera). Al final se redondeará la clase resumiendo los conceptos más importantes.

En las clases donde discutimos ejercicios de los prácticos y obligatorios se seguirá con la misma metodología.

La evaluación final se realizará mediante la resolución de ejercicios sacados del libro de texto consistiendo en 2 obligatorios (30% el primero, y 60% el segundo). El 10% restante en la evaluación se dedicará a evaluar la participación en clase. Es importante valorar a quienes siguen el curso, siguen las discusiones, hacen preguntas y participan en las clases.

Es importante aclarar que la participación en clase es obligatoria (con un 80% mínimo de asistencia), debido a que la metodología usada es fundamental para la comprensión de los temas dictados en el curso.

El curso se aprueba si se tiene el 60% o más de puntos y un 80% de asistencia a clase.

A4) CALIDAD DE LIBRE

Este curso no adhiere a resolución del consejo sobre condición de libre.

A5) CUPOS DE LA UNIDAD CURRICULAR

Este curso no tiene cupos.

ANEXO B para las carreras Ingeniería en Computación (plan 97)

B1) ÁREA DE FORMACIÓN

Programación.

B2) UNIDADES CURRICULARES PREVIAS

Para el Curso: examen aprobado de de Programación 3 y
curso aprobado de Teoría de Lenguajes
Para el Examen: no aplica.

ANEXO B para las carreras Licenciatura en Computación

B1) ÁREA DE FORMACIÓN

Programación.

A continuación se indica la pertenencia de esta unidad a agregados de perfiles de la carrera:

<i>Perfil</i>	<i>Agregado</i>
Computación Confiable	Computación Confiable
Sistemas de Información	Asignaturas Complementarias
Inteligencia Artificial	Asignaturas Complementarias

B2) UNIDADES CURRICULARES PREVIAS

Para el Curso: examen aprobado de de Programación 3 y
curso aprobado de Teoría de Lenguajes

Para el Examen: no aplica.