



Programa de Teoría de la Computación.

1. NOMBRE DE LA UNIDAD CURRICULAR

Teoría de la Computación.

2. CRÉDITOS

7 (siete)

3. OBJETIVOS DE LA UNIDAD CURRICULAR

Que el estudiante conozca diversos modelos matemáticos de computabilidad.

Que estudie la equivalencia de algunos de estos modelos y la tesis de formalización de la noción de procedimiento mecánico sustentada en dicha equivalencia.

Que conozca la existencia de clases de problemas no resolubles por métodos mecánicos y métodos de demostración asociados.

Que tenga una noción inicial sobre las clases de complejidad de problemas, casos de problemas y su clasificación y métodos de demostración asociados.

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.

Al final del curso hay una prueba final.

Horas clases teórico-prácticas:	40 hs.
Horas de estudio para clases y parcial:	31 hs.
Horas trabajo obligatorio:	30 hs.
Horas parcial:	4 hs.
 Total:	 105 hs.

5. TEMARIO

1. Introducción (Repaso): Funciones, conjuntos numerables y no numerables, método de diagonalización.
2. Lenguaje P como modelo de computación. Sintaxis y semántica del lenguaje P. Definición de Macros dentro del lenguaje P.
3. Problemas decidibles e indecidibles. Metodología: diagonalización y reducción de problemas
4. Conjuntos recursivamente enumerables.
5. Otros modelos computacionales, Máquinas de Turing, funciones recursivas. Tesis de Church.
6. Muy breve introducción a complejidad computacional. Clases P, NP y problemas NP-Completo.

6. BIBLIOGRAFÍA

Tema	Básica	Complementaria
Introducción (Repaso): Funciones, conjuntos numerables y no numerables, método de diagonalización.	(1)	
Lenguaje P como modelo de computación.	(1)	
Problemas decidibles e indecidibles.	(1)	(2)
Conjuntos recursivamente enumerables.	(1)	
Otros modelos computacionales, Máquinas de Turing, funciones recursivas. Tesis de Church.	(1)	
Muy breve introducción a complejidad computacional. Clases P, NP y problemas NP-Completo.	(2)	

6.1 Básica

1. A.J. Kfoury, Robert N. Moll, Michael A. Arbib (1982). A programming approach to computability. Springer Verlag ISBN 0-387-90743-2. 1982.

Contiene todo el contenido del curso hasta el punto 5 del temario, inclusive. NO toca el punto 6. Si bien es un libro que tiene ya 35 años, tiene el gran beneficio de presentar los temas fundamentales de la computación (que forman su sólida base científica y que se ha mantenido incambiada desde los años '60) bajo un punto de vista pedagógico que es muy claro. Además usar un lenguaje de programación en vez de Máquinas de Turing (modelo equivalente según la tesis de Church) permite una presentación mucho más clara que la tradicional usando Máquinas de Turing. Permite además probar teoremas en relación a los programas escritos y permite una presentación pedagógica mucho más clara de conceptos que son largamente no triviales y difíciles de entender.

El libro está accesible por el portal Timbó.

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

Este libro se usa fundamentalmente para el punto 6 del temario. Tiene además capítulos iniciales dedicados a la computabilidad computacional (tocando varios de los puntos 1 a 5 del temario). Sin embargo, su presentación usa el formalismo de Máquinas de Turing, con lo cual no es la visión que se tiene en el curso. Por otro lado, la presentación que hace sobre la introducción a la complejidad que se da en el curso, es lo suficientemente clara y entendible como para ser usada en el curso.

Hay copias del libro en la Biblioteca del Pedeciba Informática.

6.2 Complementaria

7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS

7.1 Conocimientos Previos Exigidos:

Se requiere conocimiento sobre lenguajes formales y autómatas.

7.2 Conocimientos Previos Recomendados:

Principios de algoritmia

ANEXO A
Para todas las Carreras

A1) INSTITUTO

Computación.

A2) CRONOGRAMA TENTATIVO

Consiste en un cronograma de avance semanal con detalle de las horas de clase asignadas a cada tema.

Semana 1	Breve historia de los orígenes de la computación y motivación del curso (4 hs.)
Semana 2	Funciones, conjuntos numerables y no numerables, método de diagonalización (4 hs.)
Semana 3	Ejercicios relacionados con el punto 1 del temario (2 hs.) Lenguaje P como modelo de computación (2 hs.)
Semana 4	Sintaxis y semántica del lenguaje P (2 hs.) Definición de Macros dentro del lenguaje P (2 hs.)
Semana 5	Ejercicios relacionados con el punto 2 del temario y Obligatorio 1 (2 hs.) Problemas decidibles y no decidibles (2 hs.)
Semana 6	Problemas no decidibles. Metodología: diagonalización y reducción de problemas (4 hs.)
Semana 7	Conjunto recursivamente enumerables (4 hs.)
Semana 8	Ejercicios relacionados con los puntos 3 y 4 del temario. Obligatorio 2 (2 hs.) Repaso y perspectivas de todos los temas visto hasta el momento (2 hs.)
Semana 9	Otros modelos computacionales, Máquinas de Turing, funciones recursivas y Tesis de Church (2 hs.) Motivación e introducción a la complejidad. Clases P y NP (2 hs.)
Semana 10	Problemas NPCompletos (2 hs.) Consultas finales del Obligatorio 2 (2 hs.)
Semana 11	Clases de consulta para el parcial (2 - 4 hs.), computadas como parte de las horas de estudio y no de clases dictadas.
Semana 12	Parcial final (4 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 obligatorios se seguirá con la misma metodología. De ser necesario se fijará alguna clase extra especial de consultas de los ejercicios obligatorios.

La evaluación final se realizará mediante la resolución de ejercicios a ser resueltos en 2 obligatorios (15% cada uno) y una prueba final (70%).

Es importante aclarar que la participación en clase no es obligatoria, pero se recomienda su presencia debido a que la metodología usada ayuda mucho a la comprensión de los temas dictados en el curso.

El curso se aprueba si se tiene el 60% o más de puntos y si se entrega los dos obligatorios (con un mínimo del 50% en cada uno de ellos).

A4) CALIDAD DE LIBRE

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

A5) CUPOS DE LA UNIDAD CURRICULAR

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 de Teoría de Lenguajes.

Para el Examen: no aplica.

ANEXO B para las carreras Ingeniería en Computación (plan 87)

B1) ÁREA DE FORMACIÓN

No corresponde.

B2) UNIDADES CURRICULARES PREVIAS

Para el Curso: previas comunes a las electivas
examen de Lógica y Programación III

Para el Examen: no aplica.

Observación: esta unidad curricular se corresponde con media electiva.