

PROGRAMA DE CURSO

Código	Nombre			
CC3102	Teoría de la Computación			
Nombre en Inglés				
Theory of Computation				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1.5	5.5
Requisitos			Carácter del Curso	
CC3001 Algoritmos y Estructuras de Datos, CC3101 Matemáticas Discretas para la Computación			Obligatorio para Licenciatura en Computación.	
Resultados de Aprendizaje				
Al término del curso se espera que el estudiante comprenda los conceptos más relevantes relacionados con (i) lenguajes regulares y autómatas finitos, (ii) lenguajes libres del contexto y autómatas de pila, (iii) Máquinas de Turing y computabilidad, (iv) complejidad computacional y NP-completitud.				

Metodología Docente	Evaluación General
<p>Clases expositivas del profesor de cátedra, buscando la participación de los alumnos en pequeños problemas que se van proponiendo durante la exposición.</p> <p>Clases auxiliares dedicadas a explicar ejemplos más extensos, resolver ejercicios propuestos, y preparación pre y post controles.</p>	<p>Tres controles evaluando ejercicios simples y un examen final evaluando lo aprendido con más perspectiva. NC = promedio ponderado del examen (40%) y del promedio de los controles (60%).</p> <p>Tareas tanto teóricas (resolución de ejercicios) como prácticas, que permiten al alumno comprender la conexión con las aplicaciones prácticas.</p>

Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas	
1	Lenguajes Regulares	3	
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía	
<ol style="list-style-type: none"> Alfabetos, cadenas y lenguajes. Representación finita del lenguaje. Autómatas finitos determinísticos. Autómatas finitos no determinísticos. Equivalencia entre ambos tipos de autómatas. Expresiones regulares. Equivalencia entre expresiones regulares y autómatas. Lema del bombeo. 	<p>Al término de la unidad, el alumno comprende lo que es un lenguaje regular y domina sus representaciones en forma de autómatas finitos y expresiones regulares. Conoce las formas más elementales de demostrar que un lenguaje no es regular. Comprende las aplicaciones prácticas de los autómatas a búsqueda en texto.</p>	<ol style="list-style-type: none"> Cap. 2 Cap. 1 y 2 (2.1 – 2.4) Cap. 1 	

Número	Nombre de la Unidad	Duración en Semanas	
2	Lenguajes Libres del Contexto	3	
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía	
<ol style="list-style-type: none"> Gramáticas libres del contexto. Autómatas de pila. Equivalencia entre gramáticas y autómatas. Propiedades de clausura, algorítmicas y de periodicidad. Teorema de bombeo. Determinismo y parsing. 	<p>Al término de la unidad, el alumno comprende lo que es un lenguaje libre del contexto y dominar sus representaciones en forma de autómatas de la pila y gramáticas libres del contexto. Conoce las formas más elementales de demostrar que un lenguaje no es libre del contexto. Comprende las aplicaciones prácticas de esta teoría al parsing.</p>	<ol style="list-style-type: none"> Cap. 3 Cap. 3 (3.7 sólo parte) Cap. 2 	

Número	Nombre de la Unidad	Duración en Semanas
3	Máquinas de Turing	2.5
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ol style="list-style-type: none"> Definición de Máquinas de Turing (MTs). Configuraciones y modelo de computación. Modularización y solución de problemas más complejos usando MTs. Uso de MTs para decidir lenguajes, aceptar lenguajes, y calcular funciones. Extensiones de MTs: múltiples cintas y otras. Simulaciones. MTs no determinísticas y su simulación. 	<p>Al término de la unidad, el alumno comprende el formalismo de Máquina de Turing y utilizarlo como un modelo de computación. Comprende las nociones de lenguajes decidibles y aceptables.</p>	<p>[1] Cap. 4 [2] Cap. 4 (4.1 – 4.3 y 4.5) [4] Cap. 3</p>

Número	Nombre de la Unidad	Duración en Semanas
4	Computabilidad	3.5
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ol style="list-style-type: none"> La Máquina Universal de Turing. Simulación. La Tesis de Church y otros formalismos. Equivalencia con el modelo RAM. El problema de la detención. Lenguaje decidibles y aceptables. Otros problemas indecidibles acerca de MTs. Reducción de problemas. Gramáticas dependientes del contexto. Equivalencia con MTs. Enumerabilidad. Problemas indecidibles acerca de gramáticas. Otros problemas indecidibles. 	<p>Al término de la unidad, el alumno comprende que existen problemas que no se pueden resolver por computador, la diferencia entre lenguajes decidibles y aceptables, y conoce algunos ejemplos emblemáticos, en particular el problema de la detención. Comprende la Tesis de Church y noción de algunos formalismos alternativos.</p>	<p>[1] Cap. 5 [2] Cap. 4.6 y 5 [4] Cap. 4 y 5</p>

Número	Nombre de la Unidad	Duración en Semanas
5	Complejidad Computacional	3
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ol style="list-style-type: none"> Longitud de una computación. Lenguajes y problemas. Complejidad de un problema. Abstracciones. Notación O. Las clases P y NP. Reducción polinomial. NP-completitud El problema NP-completo de satisfactibilidad de fórmulas booleanas. Otros problemas NP-completos: clique, recubrimiento de vértices circuito hamiltoniano, coloreo y otros. 	<p>Al término de la unidad, el alumno comprende el concepto de complejidad computacional y el significado de las clases de problema P y NP. Comprende el concepto de NP-completitud y conoce varios de los problemas NP-completos más famosos. Comprende el mecanismo para establecer que un problema es NP-completo.</p>	<p>[1] Cap. 6 [2] Cap. 6 y 7.1 [3] Cap. 34 [4] Cap. 7</p>

Bibliografía
<p>[1] G. Navarro. Fundamentos de la Ciencia de la Computación. Apuntes y Ejercicios, 2006 www.dcc.uchile.cl/gnavarro/apunte.html</p> <p>[2] H. Lewis, C. Papadimitriou. Elements of the Theory of Computation, 2da. Edition. Prentice-Hall, 1998.</p> <p>[3] T. Cormen. C. Leiserson, R. Rivest, C. Stein. Introduction to Algorithms, 2da. Edition. MIT Press, 2001.</p> <p>[4] M. Sipser. Introduction to the Theory of Computation, Second Edition. Thompson Course Technology, 2006.</p> <p>[5] D. Kelley. Teoría de Autómatas y Lenguajes Formales. Prentice-Hall, 1995.</p> <p>[6] J. Hopcroft, R. Motwani, J. Ullman. Introduction to Automata Theory, Languages and Computation, 2da. Edition. Pearson Education, 2001.</p> <p>[7] A. Aho, J. Hopcroft, J. Ullman. The design and Analysis of Computer Algorithms. Addison-Wesley, 1974.</p>

Vigencia desde:	Primavera 2009
Elaborado por:	Gonzalo Navarro
Revisado por:	ADD (noviembre 2009)