

### PROGRAMA DE CURSO

Código	Nombre			
CC5110	Introducción a la Complejidad Computacional			
Nombre en Inglés				
Introduction to Computational Complexity				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3		7
Requisitos			Carácter del Curso	
CC3102 Teoría de la Computación			Ramo electivo de ICC	
Resultados de Aprendizaje				
<p>Al finalizar el curso el alumno será capaz de entender los resultados fundamentales del área de la teoría de complejidad computacional, así como varias de sus consecuencias para el desarrollo de nuestra disciplina.</p>				

Metodología Docente	Evaluación General
Clases expositivas de 90 minutos cada una.	<p>La evaluación se basa en 4 tareas y un examen final.</p> <p>La nota final NF se calcula como:</p> $NF = 0,6 * NT + 0,4 * NE$ <p>donde NT es el promedio de las 4 tareas y NE es la nota del examen.</p>

### Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas
1	Máquinas de Turing	1,5
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Máquinas de Turing: Deterministas, no deterministas. Con varias cintas.	Comprender el modelo de computación fundamental de nuestra área, así como sus diferentes formulaciones.	

Número	Nombre de la Unidad	Duración en Semanas
2	Clases de Complejidad	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Nociones de complejidad computacional: Tiempo y espacio.  Separaciones básicas entre clases de complejidad.  Teoremas fundamentales sobre clases de complejidad: Savitch e Immerman- Szelepcseny.	Comprender las nociones básicas de complejidad computacional, en base a sus dos parámetros fundamentales: El tiempo y el espacio.  Entender las relaciones básicas existentes entre las clases de complejidad.	

Número	Nombre de la Unidad	Duración en Semanas
3	Reducciones y Problemas Completos	3
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Reducciones y completitud.  Problemas completos para las clases principales: PTIME, NP, PSPACE, EXPTIME.	Entender la noción de completitud para una clase de complejidad, los diferentes tipos de reducciones.  Conocer distintos tipos de problemas completos para cada clase, así como las técnicas para demostrar esto.	

Número	Nombre de la Unidad	Duración en Semanas
4	Jerarquía Polinomial	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
La jerarquía polinomial: Definición, estructura, propiedades.	Entender la función de la jerarquía polinomial en la teoría de complejidad computacional, así como su relación con las clases antes estudiadas.	

Número	Nombre de la Unidad	Duración en Semanas
5	Clases de Complejidad Paralelas	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Clases de complejidad basadas en circuitos, su estructura y su relación con las clases antes estudiadas.	Entender la utilización de circuitos como forma de modelar la paralelización de los problemas. Comprender la estructura de estas clases y su relación con otras antes vistas.	

Número	Nombre de la Unidad	Duración en Semanas
6	Tópicos Avanzados	4.5
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Complejidad aproximada, probabilística, paramétrica, de conteo, etc.	Estudiar algunos de los temas modernos de la teoría de complejidad y entender su relación con los temas antes estudiados.	

### Bibliografía

- 1- [AB09] S. Arora, B. Barak. Computational Complexity: A Modern Approach. Cambridge University Press, 2009.
- 2- [Koz05] D. Kozen. Theory of Computation. Springer, 2005.
- 3- [Pap95] Ch. Papadimitriou. Computational Complexity. Addison-Wesley, 1995.

Vigencia desde:	Primavera 2012
Elaborado por:	Pablo Barceló