

Unidad de Aprendizaje:		Cálculo lambda y tipos		
Periodo lectivo	Horas totales	Horas Teóricas	Horas Prácticas	Créditos
Segundo	4	4	0	8
Área:	Especialización			
Unidades de Aprendizaje Antecedentes		Unidades de Aprendizaje Consecuentes		
Ninguna		Ninguna		
Fecha de elaboración: Enero 2016		Elaboró: Dr. J. A. Hernández Servín		
Objetivo general: Comprender y aplicar las herramientas matemáticas para la descripción de los lenguajes modernos de programación. Conocer y entender la naturaleza lógica de la teoría de los lenguajes de programación.				
Contenido temático: Unidad I Términos Unidad II Teorías formales lambda beta y lambda beta n Unidad III Numerales de Church Unidad IV Funciones totalmente recursivas en el cálculo lambda, Unidad V Segundo teorema de recursividad y decidibilidad Unidad VI Álgebras combinatorias Unidad VII Tipos Unidad VIII Unificación Unidad IX Corrección de algoritmos con tipos principales.				
Actividades de aprendizaje: 1. Búsqueda de información: El alumno, bien individualmente o en equipo, buscará información sobre los temas que se estarán revisando, misma que será discutida en clase. 2. Resolución de problemas: De algunos temas se requiere la realización de ejemplos, en estos casos se dejarán problemas de práctica para su resolución en casa y se revisarán en clase. 3. Trabajo escrito: Dado un problema real, los estudiantes propondrán una estrategia de solución aplicando los conocimientos adquiridos en clase. Este se considera la parte práctica del proyecto final de curso, no obstante se irá desarrollando a lo largo del semestre. La segunda parte del proyecto final incluye un trabajo escrito en forma de reporte o artículo.				
Procedimiento de evaluación: Para la evaluación se realizará de acuerdo con el Capítulo VII del Reglamento de Estudios Avanzados. Se recomienda:				
		Producto de evaluación	Porcentaje	
		Examen escrito	60	
		Trabajo escrito	40	
Bibliografía [1] H. P Barendregt, <i>The lambda calculus: Its syntax and semantics</i> . Rev. ed. Amsterdam, AM, HOL: North-Holland, 1984, vol. 103. [2] M. Atkinson, P. Buneman, and R Morrison, <i>Data types and persistence</i> . Berlin, BER, DEU: Springer-Verlag, 1988. [3] J. R. Hindley, <i>Basic simple type theory</i> . Cambridge, CAM, UK.: Cambridge University Press, 1997, vol. 42. [4] B. C. Pierce, <i>Types and programming languages</i> . Cambridge, Mass, USA: MIT Press, 2002. [5] H. P Barendregt, W. Dekkers, and R. Statman, <i>Lambda calculus with types</i> . ser. Perspectives in logic. Cambridge, CAM, UK: Cambridge University Press, 2013. [6] G. Michaelson, <i>An introduction to functional programming through Lambda calculus</i> . Dover ed, ser. Dover books on mathematics. Mineola, N.Y, USA: Dover Publications, 2011.				