

PLAN DE ESTUDIOS (PE): Licenciatura en Ciencias de la Computación

ÁREA: Ciencias de la Computación

ASIGNATURA: Computabilidad

CÓDIGO: CCOM-256

CRÉDITOS: 5

FECHA: 28 de junio 2012



1. DATOS GENERALES

Nivel Educativo:	<i>Licenciatura</i>
Nombre del Plan de Estudios:	<i>Licenciatura en Ciencias de la Computación</i>
Modalidad Académica:	<i>Presencial</i>
Nombre de la Asignatura:	<i>Computabilidad</i>
Ubicación:	<i>Nivel Formativo</i>
Correlación:	
Asignaturas Precedentes:	<i>Lenguajes Formales y Autómatas</i>
Asignaturas Consecuentes:	<i>N/A</i>
Conocimientos, habilidades, actitudes y valores previos:	<p><i>Conocimientos de: lenguajes formales, lógica matemática, relaciones y funciones.</i></p> <p><i>Habilidades para: analizar y generar modelos matemáticos que impliquen soluciones a problemas computacionales.</i></p> <p><i>Actitudes y valores de: honestidad, responsabilidad, participación, respeto, adaptación, comprensión, tolerancia.</i></p>

2. CARGA HORARIA DEL ESTUDIANTE

Concepto	Horas por periodo		Total de horas por periodo	Número de créditos
	Teoría	Práctica		
Horas teoría y práctica (16 horas = 1 crédito)	80	0	80	5
Total	80	0	80	5



3. REVISIONES Y ACTUALIZACIONES

Autores:	<i>Claudia Zepeda Cortés, Alba Maribel Sánchez Gálvez, Meliza Contreras González, Mireya Tovar Vidal, Darnes Vilariño Ayala, César Bautista Ramos, Carlos Guillén Galván, Alfonso Garcés Báez, José de Jesús Lavalle Martínez.</i>
Fecha de diseño:	<i>Noviembre de 2009</i>
Fecha de la última actualización:	<i>28 de junio de 2012</i>
Fecha de aprobación por parte de la academia de área	<i>12 de julio de 2012</i>
Fecha de aprobación por parte de CDESCUA	<i>07 de febrero de 2013</i>
Fecha de revisión del Secretario Académico	<i>18 de febrero de 2013</i>
Revisores:	<i>Claudia Zepeda Cortés, Alba Maribel Sánchez Gálvez, Meliza Contreras González, Mireya Tovar Vidal, Darnes Vilariño Ayala, César Bautista Ramos, Carlos Guillén Galván, Guillermo De Ita Luna, Fernando Zacarías Flores, José de Jesús Lavalle Martínez.</i>
Sinopsis de la revisión y/o actualización:	<i>Actualización de las unidades. Se incluyeron actividades de aprendizaje que se corresponden con los objetivos específicos y los ejes transversales.</i>

4. PERFIL DESEABLE DEL PROFESOR (A) PARA IMPARTIR LA ASIGNATURA:

Disciplina profesional:	<i>Ciencias de la Computación o áreas afines</i>
Nivel académico:	<i>Posgrado en Ciencias</i>
Experiencia docente:	<i>Mínima de 2 años</i>
Experiencia profesional:	<i>Mínima de 1 año</i>

5. OBJETIVOS:

5.1 General: Analizar la teoría de las funciones recursivas para determinar si la solución de un problema es un procedimiento efectivo (decidible, indecidible o parcialmente decidible).

5.2 Específicos:

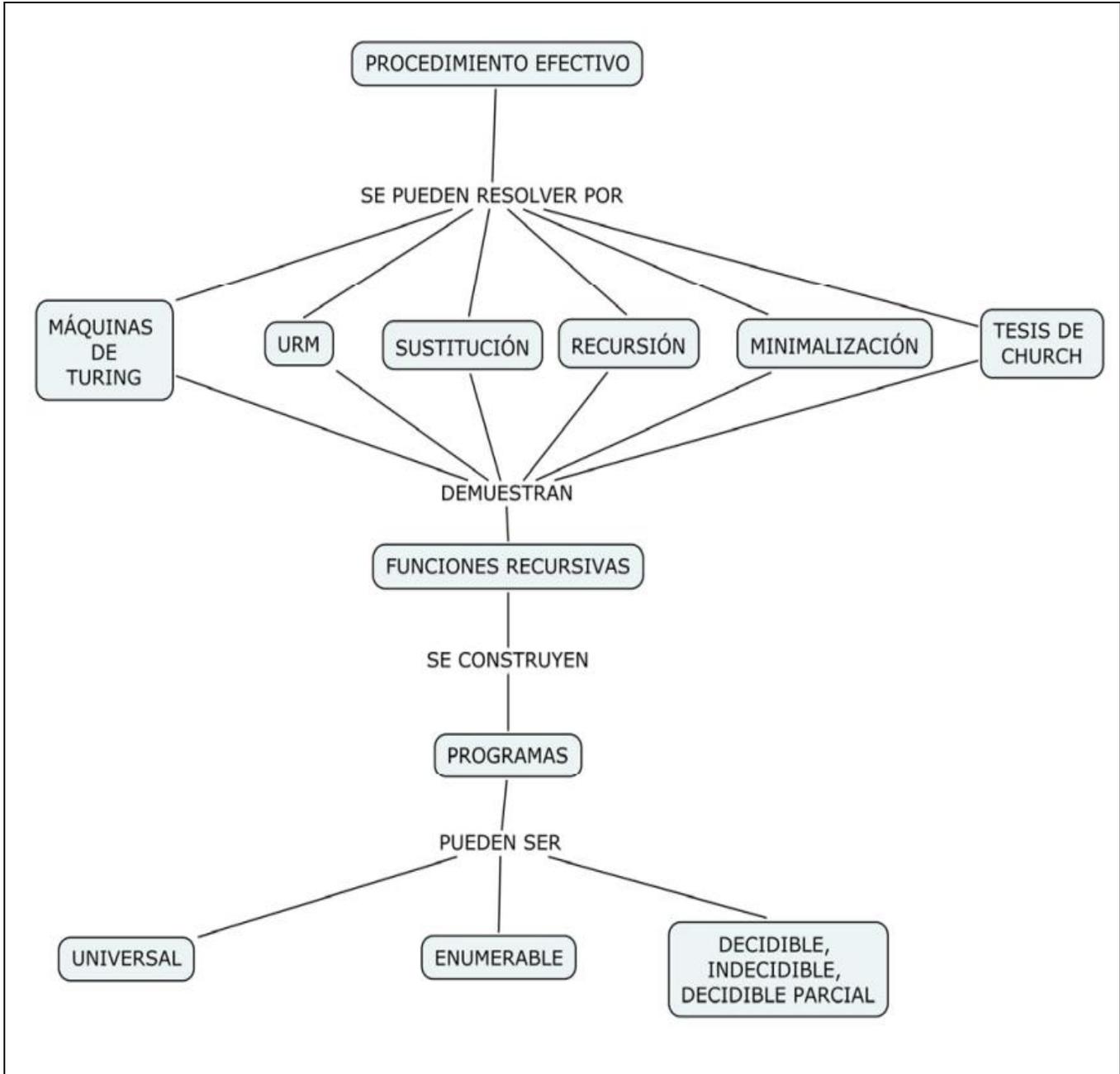
- Reconocer la representación de lenguajes y funciones en una máquina de Turing.
- Analizar el concepto de algoritmo o procedimiento efectivo.



- Analizar los elementos esenciales de la teoría de la Computabilidad y diseñar programas URM computables.
- Determinar si una función es URM computable.
- Determinar cuando un predicado es decidable.
- Extender la definición de computabilidad a otros dominios.
- Examinar los diferentes modelos para obtener nuevas funciones recursivas a partir de las funciones primitivas.
- Desarrollar programas que involucren el llamado a subrutinas.
- Diseñar funciones computables a partir de la operación de sustitución.
- Ejemplificar una gama de funciones que poseen la característica de ser recursiva.
- Generar funciones computables utilizando la operación de minimización.
- Relacionar diferentes modelos para expresar la noción de función computable.
- Contrastar el concepto de computabilidad con el de URM-computabilidad y el concepto de programa con el de URM-programa.
- Verificar funciones computables por medio de la numeración.
- Construir funciones totales no computables.
- Encontrar índices de funciones computables.
- Demostrar que la clase de funciones recursivas es enumerable.
- Relacionar funciones universales con programas universales.
- Construir funciones no computables.
- Construir con un programa universal una función total computable que no es primitiva recursiva.
- Analizar la computabilidad de la función Gamma.
- Categorizar los problemas de tipo decidable contra los de tipo indecidible para identificar los límites de la computabilidad y con ello demostrar los límites teóricos del poder de las computadoras reales.
- Utilizar el algoritmo de Sturn para probar decidibilidad, indecidibilidad y decidibilidad parcial.
- Demostrar la existencia de predicados parcialmente decibles



6. REPRESENTACIÓN GRÁFICA DE LA ASIGNATURA:



7. CONTENIDO

Unidad	Objetivo Específico	Contenido Temático/Actividades de aprendizaje	Bibliografía	
			Básica	Complementaria
1 Máquinas de Turing	Reconocer la representación de lenguajes y funciones en una máquina de Turing.	1.1 Máquina universal de Turing y el problema del paro/Ejemplifica el concepto de máquina universal de Turing y sus consecuencias con respecto a la insolubilidad.	John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman <i>Introduction to Automata Theory, Languages and Computation.</i> Addison Wesley. 3ra. Edición, 2007 Herbert B. Enderton. <i>Computability Theory: An Introduction to Recursion Theory.</i> Academic Press, 2010	Minsky M.L. <i>Computation: Finite and Infinite Machines,</i> Prentice Hall, 1967. Rogers H. Jr., <i>Theory of Recursive Functions and Effective Computability,</i> Mc. Graw Hill, 2002. Davis Martin <i>Computability and unsolvability</i> Mcgraw-Hill, 1982.

Unidad	Objetivo Específico	Contenido Temático/Actividades de aprendizaje	Bibliografía	
			Básica	Complementaria
2 Funciones Computables	Analizar el concepto de algoritmo o procedimiento efectivo	2.1 Algoritmos o Procedimientos Efectivos/Relaciona el término de función computable con el de algoritmo.	Cutland N. J. <i>Computability: An Introduction to Recursive Function Theory,</i>	Davis Martin <i>Computability and unsolvability</i> Mcgraw-Hill. 1982.



Unidad	Objetivo Específico	Contenido Temático/Actividades de aprendizaje	Bibliografía	
			Básica	Complementaria
	Analizar los elementos esenciales de la teoría de la Computabilidad y diseñar programas URM computables	2.2 La Máquina de Registros Ilimitados/Diseña programas URM.	Cambridge University Press. 1980.	
	Determinar si una función es URM computable	2.3 Funciones URM-Computables/Distingue cuando ciertas funciones son computables.	Jones, Neil D. <i>Computability and complexity From a Programming Perspective</i> , Massachusetts Institute of Technology 1997.	
	Determinar cuando un predicado es decidible	2.4 Predicados y Problemas Decidibles/ Determina cuando un predicado es decidible.	Herbert B. Enderton. <i>Computability Theory: An Introduction to Recursion Theory</i> . Academic Press, 2010.	
	Extender la definición de computabilidad a otros dominios	2.5 Computabilidad en otros dominios/Extiende la definición de computabilidad a otros dominios.		

Unidad	Objetivo Específico	Contenido Temático/Actividades de aprendizaje	Bibliografía	
			Básica	Complementaria
3 Generando funciones computables	Examinar los diferentes modelos para obtener nuevas funciones recursivas a partir de las funciones primitivas.	3.1 Las Funciones Básicas/Genera funciones computables a partir de las funciones primitivas.	Cutland N. J. <i>Computability: An Introduction to Recursive Function Theory</i> , Cambridge University Press. 1980.	Boolos George, Burgess John P. and Jeffrey Richard C. <i>Computability and logic</i> . Edición: 5 Cambridge University Press. 2007

Unidad	Objetivo Específico	Contenido Temático/Actividades de aprendizaje	Bibliografía			
			Básica	Complementaria		
	Desarrollar programas que involucren el llamado a subrutinas.	3.2 Unión de programas/Desarrolla programas que incluyan a otros programas como subprogramas o subrutinas.	Herbert B. Enderton. <i>Computability Theory: An Introduction to Recursion Theory.</i> Academic Press, 2010.	Davis Martin. <i>Computability and unsolvability</i> Mcgraw-Hill. 1982.		
	Diseñar funciones computables a partir de la operación de sustitución.	3.3 Substitución/Demuestra que una colección de funciones es computable utilizando las operaciones de sustitución.				
	Ejemplificar una gama de funciones que poseen la característica de ser recursiva.	3.4 Recursión/Demuestra que una colección de funciones es computable utilizando las operaciones de recursión.				
	Generar funciones computables utilizando la operación de minimización.	3.5 Minimalización/Genera funciones computables a partir de la minimización.				
	Relacionar diferentes modelos para expresar la noción de función computable.	3.6 Tesis de Church / Relaciona el modelo de la máquina de registros ilimitados con el modelo de funciones recursivas parciales.				
Unidad	Objetivo Específico	Contenido Temático/Actividades de aprendizaje	Bibliografía			
Básica	Complementaria	4 Numerando funciones computables	Contrastar el concepto de computabilidad con el de URM-computabilidad y el concepto de programa con el	4.1 Numerando Programas/Relaciona el concepto de programa computable con el de función por medio de una biyección con los números naturales.	Cutland N. J. <i>Computability: An Introduction to Recursive Function Theory,</i> Cambridge	Boolos George, Burgess John P. and Jeffrey Richard C. <i>Computability and logic.</i> Edición: 5

Unidad	Objetivo Específico	Contenido Temático/Actividades de aprendizaje	Bibliografía	
			Básica	Complementaria
	de URM-programa.		University Press. 1980.	Cambridge University Press. 2007
	Verificar funciones computables por medio de la numeración.	4.2. Numerando Funciones Computables/Verifica funciones computables.		Davis Martin. <i>Computability and unsolvability</i> Mcgraw-Hill. 1982.
	Construir funciones totales no computables.	4.3. El Método Diagonal /Construye funciones totales, pero no computables.		
	Encontrar índices de funciones computables.	4.4. El Teorema s-m-n/Encuentra el índice de funciones computables.		
	Demostrar que la clase de funciones recursivas es enumerable.	4.5. Conjuntos recursivos/Demuestra que la clase de funciones recursivas es enumerable.		

Unidad	Objetivo Específico	Contenido Temático/Actividades de aprendizaje	Bibliografía	
			Básica	Complementaria
5 Programas universales	Relacionar funciones universales con programas universales.	5.1 Funciones Universales y Programas Universales/Relaciona una función universal con un programa universal.	Cutland N. J. <i>Computability: An Introduction to Recursive Function Theory</i> , Cambridge University Press. 1980. Herbert B. Enderton. <i>Computability Theory: An Introduction to</i>	
	Construir funciones no computables.	5.2 Aplicaciones del programa universal/Construye funciones no computables específicas a partir de programas universales/Construye predicados indecidibles a partir de programas universales.		
	Construir con un	5.3.Operaciones		



Unidad	Objetivo Específico	Contenido Temático/Actividades de aprendizaje	Bibliografía	
			Básica	Complementaria
	programa universal una función total computable que no es primitiva recursiva.	Efectivas en Funciones Computables/Relaciona el teorema s-m-n con el de programa universal.	<i>Recursion Theory.</i> Academic Press, 2010.	
	Analizar la computabilidad de la función Gamma.	5.4. Computabilidad de la Función GAMMA (n)/Resuelve ejercicios.		

Unidad	Objetivo Específico	Contenido Temático/Actividades de aprendizaje	Bibliografía	
			Básica	Complementaria
6 Decidibilidad, indecidibilidad y decidibilidad parcial	Categorizar los problemas de tipo decidible contra los de tipo indecidible para identificar los límites de la computabilidad y con ello demostrar los límites teóricos del poder de las computadoras reales.	6.1. Problemas Indecidibles en Computabilidad/ Identifica los tipos de problemas en teoría de la computabilidad que son indecidibles y decidibles	Cutland N. J. <i>Computability: An Introduction to Recursive Function Theory,</i> Cambridge University Press. 1980.	Davis Martin. <i>Computability and unsolvability</i> Mcgraw-Hill. 1982.
	Utilizar el algoritmo de Sturm para probar decidibilidad, indecidibilidad y decidibilidad parcial	6.2. El Algoritmo de Sturm/Utiliza el algoritmo de Sturm para probar decidibilidad, indecidibilidad y decidibilidad parcial	Herbert B. Enderton. <i>Computability Theory: An Introduction to Recursion Theory.</i> Academic Press, 2010.	



Unidad	Objetivo Específico	Contenido Temático/Actividades de aprendizaje	Bibliografía	
			Básica	Complementaria
	Demostrar la existencia de predicados parcialmente decidibles	6.3. Predicados parcialmente decidibles/Demuestra que existen predicados parcialmente decidibles		

8. CONTRIBUCIÓN DEL PROGRAMA DE ASIGNATURA AL PERFIL DE EGRESO

Asignatura	Perfil de egreso		
	Conocimientos	Habilidades	Actitudes y valores
Computabilidad	En los fundamentos Matemáticos de la Ciencia de la computación, En los conceptos principales y las teorías relacionadas con la Ciencia de la Computación, a través de la construcción de máquinas abstractas, de la modelación y representación de entidades o sistemas en diferentes niveles de abstracción.	En el análisis y generación de modelos matemáticos que impliquen soluciones a problemas computacionales a partir del desarrollo de la capacidad de análisis y síntesis, de la creatividad en el diseño de modelos matemáticos para un problema específico.	Mostrará una actitud positiva y favorable a los cambios científicos. Estará preparado para incorporarse en el marco de la globalización. Estará preparado para trabajar en equipo, emprender, liderar proyectos e incidir en la transformación sustentable de la realidad. Será un profesional responsable, solidario, crítico, ético y comprometido con la sociedad.



9. Describa cómo el eje o los ejes transversales contribuyen al desarrollo de la asignatura

Eje (s) transversales	Contribución con la asignatura
Formación Humana y Social	Fomentando el trabajo en equipos por medio de tareas colectivas. Fomentando la responsabilidad colectiva.
Desarrollo de Habilidades en el uso de las Tecnologías de la Información y la Comunicación	Haciendo búsquedas de información en Internet, para acceder a los últimos resultados obtenidos en esta línea de investigación.
Desarrollo de Habilidades del Pensamiento Complejo	Resolviendo nuevos problemas a través de la creación de modelos matemáticos.
Lengua Extranjera	Usando bibliografía en idioma inglés consolidando el conocimiento de esta lengua.
Educación para la Investigación	Resolviendo tareas que involucran búsquedas, planeamiento, corroboración o falsación de hipótesis.

10. ORIENTACIÓN DIDÁCTICO-PEDAGÓGICA.

Estrategias y Técnicas de aprendizaje-enseñanza	Recursos didácticos
<p>Estrategias de aprendizaje: Lectura y comprensión, reflexión, comparación, resumen.</p> <p>Estrategias de enseñanza: Aprendizaje activo, aprendizaje cooperativo, aprendizaje colaborativo y basado en el descubrimiento.</p> <p>Ambientes de aprendizaje: Aula, biblioteca y simuladores.</p> <p>Actividades y experiencias de aprendizaje: Analiza, diseña, construye, demuestra, programa y categoriza.</p>	<p>Materiales: Proyectors, TICs, plumón y pizarrón.</p>

11. CRITERIOS DE EVALUACIÓN

Criterios	Porcentaje
▪ Exámenes	40%
▪ Participación en clase	10%
▪ Tareas	20%
▪ Exposiciones	10%
▪ Proyecto final	20%
Total	100%



12. REQUISITOS DE ACREDITACIÓN

Estar inscrito como alumno en la Unidad Académica en la BUAP
Asistir como mínimo al 80% de las sesiones
La calificación mínima para considerar un curso acreditado será de 6
Cumplir con las actividades académicas y cargas de estudio asignadas que señale el PE

13. Anexar (copia del acta de la Academia y de la CDESCUA con el Vo. Bo. del Secretario Académico)

