



PLAN DE ESTUDIOS (PE): Licenciatura en Ingeniería en Tecnologías de la

Información

ÁREA: Optativas

ASIGNATURA: Métodos Formales

CÓDIGO: ITIS 605

CRÉDITOS: 6

FECHA: Mayo de 2017



1. DATOS GENERALES

Nivel Educativo:	<i>Licenciatura</i>
Nombre del Plan de Estudios:	<i>Licenciatura en Ingeniería en Tecnologías de la Información</i>
Modalidad Académica:	<i>Presencial</i>
Nombre de la Asignatura:	<i>Métodos Formales</i>
Ubicación:	<i>Nivel Formativo</i>
Correlación:	
Asignaturas Precedentes:	<i>Matemáticas Discretas</i>
Asignaturas Consecuentes:	<i>Ninguna</i>

2. CARGA HORARIA DEL ESTUDIANTE

Concepto	Horas por semana		Total de horas por periodo	Total de créditos por periodo
	Teoría	Práctica		
Horas teoría y práctica <i>Actividades bajo la conducción del docente como clases teóricas, prácticas de laboratorio, talleres, cursos por internet, seminarios, etc.</i> (16 horas = 1 crédito)	<u>1</u>	<u>4</u>	<u>90</u>	<u>6</u>



3. REVISIONES Y ACTUALIZACIONES

Autores:	<i>Abraham Sánchez López</i> <i>Claudia Zepeda Cortes</i>
Fecha de diseño:	<i>Junio de 2013</i>
Fecha de la última actualización:	<i>25 de Mayo de 2017</i>
Fecha de aprobación por parte de la academia de área, departamento u otro.	
Revisores:	<i>Claudia Zepeda Cortes</i> <i>Abraham Sánchez López</i> <i>José Luis Carballido Carranza</i>
Sinopsis de la revisión y/o actualización:	<i>Ajuste completo al nuevo formato; Reajuste de la redacción del propósito del curso; Especificación de competencias profesionales.</i>

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

Disciplina profesional:	<i>Ciencias de la Computación, Tecnologías de la Información, Sistemas Computacionales.</i>
Nivel académico:	<i>Maestría</i>
Experiencia docente:	<i>Mínima de 2 años</i>
Experiencia profesional:	<i>Mínima de 2 años</i>

5. PROPÓSITO: *Garantizar la rigurosidad, consistencia y completitud del desarrollo de software y con ello evitar los problemas que son el origen de los errores en el software.*

6. COMPETENCIAS PROFESIONALES: *Esta asignatura contribuye a las siguientes competencias:*

6. *Aplica modelos matemáticos de la teoría de ecuaciones o funciones, definiendo cursos de acción con pasos específicos para el desarrollo e implementación de las Tecnologías de la Información en el control y la toma de decisiones en los ámbitos de la administración pública y privada, así como las redes sociales y de generación del conocimiento.*
7. *Realiza el análisis, diseño e implementación del desarrollo de software con la finalidad de integrar elementos de seguridad y confiabilidad en la aplicación de las Tecnologías de la Información.*
8. *Desarrolla proyectos de software en grupos multidisciplinarios de trabajo, mediante la aplicación de la capacidad crítica, de análisis y síntesis con la finalidad de*

Benemérita Universidad Autónoma de Puebla
Vicerrectoría de Docencia
Dirección General de Educación Superior
Facultad de Ciencias de la Computación



generar innovaciones tecnológicas que atiendan las problemáticas del contexto social, tecnológico, ambiental y/o cultural que lo rodean.





Justificación: Los métodos formales utilizan modelos matemáticos, que permiten garantizar la rigurosidad, consistencia y completitud del desarrollo de software; evitando problemas que dan origen a errores en el software.

7. CONTENIDOS TEMÁTICOS

Unidad de Aprendizaje	Contenido Temático	Referencias
1. Introducción a los métodos formales en ingeniería de software	1.1. Definición de métodos formales 1.2. Algunas aplicaciones 1.3. El rol de los métodos formales en ingeniería de software 1.4. Limitaciones de los métodos formales 1.5. Un panorama de los métodos formales	Gabbar, Hossam A. (2010). Modern formal methods, verification and applications, Springer-Verlag, USA. Charatan, Q., Kans, A. (2004). Formal software development: From VDM to Java, Palgrave MacMillan, USA.
2. Lenguaje OCL (Object constraint language)	2.1. Objetos, escenarios y diagramas 2.2. Introducción a OCL 2.3. Expresiones y scripts en OCL 2.4. Correspondencia y remplazo de igualdad 2.5. Operadores, variables 2.6. Ejemplos prácticos	Kyas, Marcel (2006). Verifying OCL specifications of UML models: Tool support and compositionality, Lehmanns, USA. Warner, Jos., Kleppe, A. (2003) Object constraint language: The getting your models ready for MDA, Addison-Wesley, USA.
3. Redes de Petri	3.1. Introducción 3.2. Modelización con redes de Petri (modelo básico, elementos, evolución) 3.3. Estructuras fundamentales para la modelización de sistemas (paralelismo, sincronización, compartición de recursos, etc.) 3.4. Propiedades de las redes de Petri ordinarias 3.5. Redes de Petri coloreadas 3.6. Ejemplos prácticos	Van der Aalst, Wil Van., Stahl, Christian (2011) Modeling business processes: A petri net-oriented approach, The MIT Press, USA. Diaz, Michel (Editor) (2009) Petri nets: Fundamental models, verification and applications, John Wiley & Sons, USA.
4. Autómatas	4.1. Identificar los elementos básicos de un lenguaje 4.2. Autómatas finitos deterministas y no-deterministas 4.3. Derivación y lenguaje generado por una gramática regular 4.4. Lema del bombeo 4.5. Autómata de pila y gramáticas libres de contexto 4.6. Ejemplos prácticos	Linz, Peter (2011) An introduction to formal languages and automata. Fifth Edition, Jones & Bartlett Learning, USA. Hopcroft, John E., Motwani, Rajeev, Ullman, Jeffrey D. (2006) Introduction to automata theory, languages and computation, 3 rd Edition, Prentice Hall, USA.





8. ESTRATEGIAS, TÉCNICAS Y RECURSOS DIDÁCTICOS (Enunciada de manera general para aplicarse durante todo el curso)

Estrategias y técnicas didácticas	Recursos didácticos
<p><u>Estrategias de Aprendizaje:</u></p> <ul style="list-style-type: none"> <i>El estudiante deberá leer textos, destacará conceptos, elaborará mapas conceptuales, organizará, jerarquizará y aplicará información.</i> <p><u>Estrategias de enseñanza:</u></p> <ul style="list-style-type: none"> <i>El profesor Jerarquizará la información y usará preferentemente las técnicas grupales como el aprendizaje colaborativo.</i> <p><u>Ambientes de aprendizaje:</u></p> <ul style="list-style-type: none"> <i>Disponibilidad de salones adecuados, bibliotecas y licencias del software requerido.</i> <p><u>Actividades y experiencias de aprendizaje:</u></p> <ul style="list-style-type: none"> <i>Se realizarán actividades para el uso del software requerido, también se realizarán actividades que involucren diálogo, redescubrimiento, técnicas grupales, mapas conceptuales, entre otras.</i> 	<p><u>Materiales:</u></p> <ul style="list-style-type: none"> <u>Materiales convencionales:</u> <ul style="list-style-type: none"> <i>libros y/o fotocopias</i> <u>Tableros didácticos:</u> <ul style="list-style-type: none"> <i>Pizarrón</i> <u>Nuevas tecnologías:</u> <ul style="list-style-type: none"> <i>Procesador Latex</i> <i>Visual Studio</i> <i>Lenguajes de propósito general</i> <i>LISP</i> <u>Servicios telemáticos:</u> <ul style="list-style-type: none"> <i>Sitios Web</i> <i>Moodle</i>





9. EJES TRANSVERSALES

Describe cómo se fomenta(n) el eje o los ejes transversales en la asignatura

Eje (s) transversales	Contribución con la asignatura
Formación Humana y Social	<i>Análisis, reflexión y juicio crítico para utilizar los diferentes lenguajes formales en el modelado de sistemas de software.</i>
Desarrollo de Habilidades en el uso de las Tecnologías de la Información y la Comunicación	<i>Búsqueda de información electrónica relacionada con el desarrollo formal de sistemas de software.</i>
Desarrollo de Habilidades del Pensamiento Complejo	<i>Comprensión del uso de métodos formales en el desarrollo de aplicaciones, como el único medio para el desarrollo de aplicaciones críticas.</i>
Lengua Extranjera	<i>Facilita la comunicación del conocimiento en otros idiomas y además la comprensión de textos científicos.</i>
Innovación y Talento Universitario	<i>Este curso aporta los elementos diferenciadores en el desarrollo de aplicaciones críticas, lo que contribuye a una mayor competitividad en el desarrollo de software moderno.</i>
Educación para la Investigación	<i>Habilidad para descubrir y construir nuevos conocimientos aplicables a la solución de problemas planteados en las tecnologías de la información.</i>

10. CRITERIOS DE EVALUACIÓN *(de los siguientes criterios propuestos elegir o agregar los que considere pertinentes utilizar para evaluar la asignatura y eliminar aquellos que no utilice, el total será el 100%)*

Criterios	Porcentaje
▪ <i>Exámenes</i>	40%
▪ <i>Participación en clase</i>	10%
▪ <i>Tareas</i>	20%
▪ <i>Proyecto final</i>	30%
Total	100%

11. 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.
No se debe contar con antecedentes comprobados de copia o plagio de prácticas o proyectos durante el curso.
Cumplir con las actividades académicas y cargas de estudio asignadas que señale el PE

Notas:

Benemérita Universidad Autónoma de Puebla
Vicerrectoría de Docencia
Dirección General de Educación Superior
Facultad de Ciencias de la Computación



- a) La entrega del programa de asignatura con sus respectivas actas de aprobación, deberá realizarse en formato electrónico, vía oficio emitido por la Dirección o Secretaría Académica a la Dirección General de Educación Superior.
- b) La planeación didáctica deberá ser entregada a la coordinación de la licenciatura en los tiempos y formas acordados por la Unidad Académica.

