

1. DATOS DE LA ASIGNATURA

Nombre de la asignatura:	Mantenimiento de Software
Carrera:	Ing. en Sistemas Computacionales
Clave de la asignatura:	ISC-2105
SATCA	2-2-4

2. HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Culiacán, 30 de abril de 2021.	<ul style="list-style-type: none">• Dr. Ricardo Rafael Quintero Meza• Dr. Clemente García Gerardo• MC. Martha Valenzuela Tirado	Elaboración de asignatura: Mantenimiento de Software.

3. PRESENTACIÓN

Caracterización de la asignatura

La materia de Mantenimiento de software forma parte de la especialidad de Ingeniería de Software de la carrera de Ingeniería en Sistemas Computacionales. Esta materia sucede a Fundamentos de Ingeniería del Software. Se recomienda su impartición en el octavo o noveno semestre.

Cuando el alumno llega a este curso ha adquirido las competencias fundamentales para la gestión de las diferentes disciplinas del ciclo de vida de software. En el curso de Fundamentos de Ingeniería de Software adquirió competencias relativas a los aspectos ingenieriles considerando la construcción de un sistema software desde su concepción, pasando por su construcción y liberación final.

Una vez que el sistema es liberado por primera vez pasa a una etapa mayor de tiempo donde requiere evolucionar para adaptarse a las nuevas necesidades que demanda el mundo real, así como también donde suelen surgir posibles fallas (no detectadas en principio) que es necesario corregir. Por estas razones se requiere de un profesional que posea las competencias que le permitan atender a estas

necesidades de mantenimiento, diferentes a las requeridas para atender el ciclo de vida inicial de la Ingeniería de Software.

Esta materia suma al perfil profesional del alumno las competencias necesarias para que el estudiante sea capaz de gestionar la evolución de sistemas software aplicando diversas técnicas de mantenimiento, con lo cual logrará complementar sus habilidades para la gestión completa el ciclo de vida del sistema: desde su concepción hasta su eliminación.

Intención didáctica

Esta materia tiene un carácter teórico-práctico en el cual el maestro se constituirá en el facilitador principal para el desarrollo de las diversas competencias generales, específicas y genéricas en el alumno. La materia es novedosa respecto a otras versiones de la especialidad de Ingeniería de Software por lo que requerirá del maestro una especial preparación teórico-práctica en los temas, así como en la creación, organización, diseño y ejecución de estrategias didácticas adecuadas que permitan el desarrollo de las competencias propuestas en el alumno.

La materia se organiza en cuatro unidades que pretenden cubrir las competencias básicas que requiere poseer un Ingeniero de Software para abordar la tarea de Mantenimiento y Evolución del Software. Su intención didáctica se describe a continuación:

En la Unidad 1 se abordan los conceptos básicos del área de Mantenimiento del Software, así como el marco de trabajo típico que caracteriza la disciplina. En esta unidad el alumno desarrollará competencias específicas relativas al saber, capacidad de trabajo en equipo, habilidades interpersonales y compromiso ético.

En la Unidad 2 se abordan las técnicas básicas de mantenimiento de software: comprensión de programas, ingeniería inversa, reestructuración y reingeniería. En esta unidad el alumno desarrollará competencias relativas al saber y el saber hacer. El profesor puede dar énfasis al desarrollo de competencias en el saber hacer principalmente, mediante la motivación al aprendizaje haciendo uso de las técnicas y herramientas que típicamente soportan el mantenimiento de software. Respecto al saber, éste se puede ir desarrollando conforme se exploran las técnicas y herramientas subyacentes. Las competencias genéricas que se pretenden desarrollar serían: capacidad de planificar y organizar, solución de problemas, capacidad de trabajar en equipo, habilidades interpersonales, capacidad de aplicar conocimientos los conocimientos en la práctica y liderazgo.

En la Unidad 3 se aborda el tema de la Gestión de Configuración. Las competencias que el alumno debe desarrollar en esta unidad se apoyan en

aquéllas adquiridas en la Unidad 2, agregando las técnicas y herramientas que demanda esta área de estudio. La unión de técnicas de mantenimiento de software y la gestión de configuración ofrecen un marco de trabajo completo para resolver el problema de la evolución del software. Por lo anterior el profesor deberá diseñar estrategias didácticas que permitan al alumno desarrollar el saber y el saber hacer buscando integrar las competencias de las unidades 2 y 3. Las competencias específicas son semejantes a las de la Unidad 2, así como también las genéricas.

Finalmente, la Unidad 4 aborda el tema de la medición del trabajo de mantenimiento y evolución a través de las métricas. La intención didáctica subyacente es que el alumno desarrollar competencias relativas al saber, el saber hacer y el saber transferir: conocer las métricas más importantes para evaluar el desempeño del trabajo de mantenimiento y con ello contar con elementos de mejora para esta tarea en el equipo de Ingeniería de Software.

4. COMPETENCIAS A DESARROLLAR:

Competencia general:

Modificar el software por medio de procesos definidos y disciplinados de gestión de configuración para proporcionar un soporte rentable antes y después de su entrega.

Competencias específicas:

- Identificar los conceptos básicos del mantenimiento de software.
- Desarrollar el mantenimiento de software aplicando diversas técnicas y herramientas: comprensión de programas, ingeniería inversa, reestructuración y reingeniería.
- Administrar el mantenimiento de software a través de la gestión de configuración
- Medir el rendimiento de las actividades de mantenimiento de software

Competencias genéricas:

Competencias instrumentales

- Capacidad de planificar y organizar.
- Comunicación oral y escrita.
- Solución de problemas.

Competencias interpersonales

- Capacidad de trabajar en equipo.
- Interdisciplinario.
- Habilidades interpersonales.
- Compromiso ético.

	Competencias sistémicas <ul style="list-style-type: none"> • Capacidad de aplicar los conocimientos en la práctica. • Liderazgo. • Capacidad para diseñar y gestionar mantenimiento de software.
--	--

5. COMPETENCIAS PREVIAS

De Fundamentos de Ingeniería de Software:

- Identificar la historia del software, así como las características de los paradigmas estructurado y orientado a objetos.
- Aplicar modelos, técnicas y herramientas para cada una de las etapas del ciclo de vida de desarrollo de software.

6. TEMARIO

Unidad	Temas	Subtemas
1	El Contexto del Mantenimiento	1.1 Introducción y definiciones básicas. 1.2 Diferencias entre el desarrollo y el mantenimiento. 1.3 Razones que justifican el mantenimiento. 1.4 Mantenimiento de Sistemas Efectivo 1.5 El Framework del Mantenimiento 1.6 Fundamentos del cambio del software 1.7 El Proceso del Mantenimiento (L. Texto: Caps. 1-3,5)
2	Técnicas de Mantenimiento	2.1 Introducción. 2.2 La comprensión de programas 2.3 Ingeniería Inversa 2.4 Reestructuración 2.5 Reingeniería (L. Texto: Caps. 6-7, L. Consulta –

		<i>Fowler Cap. 1)</i>
3	Gestión de Configuración	3.1 Introducción 3.2 Definiciones 3.3 Gestión de Configuración. 3.4 Control de Cambios 3.5 Documentación 3.6 Control de versiones (L. Texto: Cap. 11; Taller GIT)
4	Métricas de Mantenimiento	4.1 Introducción 4.2 Definiciones 4.3 Objetivos de las mediciones. 4.4 Ejemplos de mediciones. (L. Texto: Cap. 12)

7. SUGERENCIAS DIDÁCTICAS PARA EL DESARROLLO DE COMPETENCIAS ESPECÍFICAS

Competencias específicas	Actividades de aprendizaje
Unidad 1. El Contexto del Mantenimiento Identificar los conceptos básicos del mantenimiento de software.	<ul style="list-style-type: none"> • Elaboración de un mapa conceptual/mental que ilustre los conceptos básicos de mantenimiento. • Análisis y discusión mediante debates o mesas redondas sobre el tema: ¿Por qué es importante el mantenimiento?
Unidad 2. Técnicas de Mantenimiento Desarrollar el mantenimiento de software aplicando diversas técnicas y herramientas: comprensión de programas, ingeniería inversa, reestructuración y reingeniería.	<ul style="list-style-type: none"> • Análisis de un sistema para su comprensión mediante técnicas top-down y bottom-up. • Análisis de un sistema para construir especificaciones de un nivel mayor de abstracción partiendo del código fuente haciendo uso de herramientas de ingeniería inversa.

Unidad 3. Gestión de Configuración Administrar el mantenimiento de software a través de la gestión de configuración	<ul style="list-style-type: none"> • Uso de herramientas de Gestión de Configuración y Control de Cambios
Unidad 4. Métricas de Mantenimiento Medir el rendimiento de las actividades de mantenimiento de software	<ul style="list-style-type: none"> • Medir el desempeño del mantenimiento en algún caso práctico de aplicación de técnica de mantenimiento visto anteriormente.

8. SUGERENCIAS DIDÁCTICAS PARA EL DESARROLLO DE COMPETENCIAS GENÉRICAS.

<ul style="list-style-type: none"> • Desarrollar actividades que motiven al estudiante a la investigación documental sistemática y organizada. • Definir actividades para la presentación oral y escrita de los resultados obtenidos de los trabajos de investigación • Definir proyectos de mantenimiento de software que busquen resolver problemas reales organizados con equipos que definan roles y responsabilidades bien establecidos. Resaltar la importancia del liderazgo y la ética en el funcionamiento del equipo. • Utilizar herramientas de gestión de configuración y control de versiones para apoyar los ejercicios de mantenimiento.

9. SUGERENCIAS DE EVALUACIÓN

Diagnóstica	Formativa	Sumativa
<ul style="list-style-type: none"> • Aplicación de Cuestionario de intereses. profesionales • Realización de Lluvia de ideas enfocada a la exploración de conocimientos previos de Ingeniería de Software 	<ul style="list-style-type: none"> • Presentación de avances de solución de casos resueltos con técnicas de mantenimiento de software. 	<ul style="list-style-type: none"> • 3 exámenes parciales (60%). • Proyectos de mantenimiento, ingeniería inversa, reestructuración y reingeniería (20%) • Proyecto de Gestión de Configuración y

		Control de Versiones
--	--	----------------------

10. FUENTES DE INFORMACIÓN

- **Libro de texto:**
 - Grubb Penny, Takang Armstron A. *Software Maintenance, concepts and practice*. 2nd Ed. World Scientific. 2003.
- **Lecturas complementarias:**
 - Tripathy Priyadarshi, Naik Kshirasagar. *Software Evolution and Maintenance, a practitioner's approach*. John Wiley & Sons. 2015.
 - Fowler Martin. *Refactoring*. Addison Wesley. 1er. Edition. 2012. ISBN B007WTFWJ6
 - Sicilia Miguel Angel, Ingenieria de software *Un enfoque desde la guia SWEBOK: Alfaomega*, 1er Edicion. 2010. ISBN 9786077074205

11. PERFIL DEL PROFESOR QUE IMPARTIRÁ LA MATERIA

- Ingeniero en Sistemas Computacionales, Licenciado en Informática o carrera afín con experiencia práctica en mantenimiento de software.
- Maestro en Ciencias Computacionales.
- Doctor en Ciencias Computacionales.