



UNIVERSIDAD AUTÓNOMA DE SINALOA
SECRETARÍA ACADÉMICA UNIVERSITARIA
 Coordinación General de Evaluación, Innovación y Calidad Educativa
UNIDAD ACADÉMICA FACULTAD DE INFORMÁTICA CULIACÁN

PROGRAMA DE ESTUDIOS

1. DATOS DE IDENTIFICACIÓN			
UNIDAD DE APRENDIZAJE	Modelos de Calidad de Software		
Clave:	4837		
Horas y créditos:	Teóricas: 40	Prácticas: 40	Estudio Independiente: 20
	Total de horas: 80		Créditos: 7
Tipo de unidad de aprendizaje:	Teórico: X	Teórico-práctico :	Práctico
Competencia (s) del perfil de egreso que desarrolla o a las que aporta.	Realiza el análisis del software, que permita el desarrollo de una Producto de Software de Calidad utilizando estándares y guías.		
Cursos consecuentes relacionados:	<ul style="list-style-type: none"> • Ingeniería de software • Análisis y Diseño de software. 	Cursos subsecuentes relacionados:	<ul style="list-style-type: none"> • Sistemas Distribuidos • Practicas Profesionales.
Responsables de elaborar y/o actualizar el programa:	MC. Alfredo Rojo García		
Fecha de elaboración:	Junio 2015	Actualización: Enero del 2017	
2. PROPÓSITO			
El Alumno Comprenderá la importancia del desarrollo de productos de software que sean de calidad, siguiendo los diferentes modelos y estándares que han sido desarrollados en el área, permitiéndole desarrollar mejores productos que ayuden a la productividad del usuario.			
3. SABERES			
Teóricos:	Conocer los diferentes modelos y estándares existentes en el área desarrollo de software.		
Prácticos:	Seleccionar los modelos o estándares de calidad más adecuados a utilizar para la elaboración de un software.		

Actitudinales:	Desarrollo de un pensamiento analítico y crítico para el desarrollo de productos de software basados en estándares de Calidad.	
4. CONTENIDOS		
BLOQUE I: Introducción a Conceptos Generales.	Aprendizajes Esperados	
<ul style="list-style-type: none"> 1.1. Conceptos Generales de Software 1.2. Costo del Software <ul style="list-style-type: none"> 1.2.1. Directo 1.2.2. Indirecto 1.2.3. Oculto 1.3. Fallas del Software 1.4. Retrasos y Cancelaciones 1.5. Complejidad Del Software <ul style="list-style-type: none"> 1.5.1. Del Problema 1.5.2. De la Solución 1.6. Aspectos para el éxito de un sistema 1.7. Diagrama Riqueza Funcional vs Calidad vs Tiempo/costo 1.8. Bala de Plata 1.9. Ciclo de vida software 1.10. Mitos del Software 1.11. Modelo SEI de 5 grados 1.12. Métricas del proceso y mejoras de proceso del software 1.13. Análisis y gestión de riesgos 1.14. Análisis de Requisitos. 	<ul style="list-style-type: none"> • Recordará los conceptos básicos del software • Conocerá las consecuencias en la historia de los fallos y retrasos en el desarrollo de software. • Reconocerá los Aspectos para el éxito de un sistema. • Conocerá el modelo SEI de 5 niveles. • Comprenderá la importancia del análisis y gestión de riesgos. 	
BLOQUE II: Calidad, Productividad y Competitividad		
<ul style="list-style-type: none"> 2.1. Elementos Significativos para la competitividad de una empresa. 2.2. Recursos en la empresa. 2.3. Importancia de la productividad 2.4. Calidad en los sistemas de servicios 2.5. La economía de la calidad 2.6. Ventajas de cuantificar los costos de calidad. 2.7. Clasificación de los costos de calidad 	<ul style="list-style-type: none"> • Comprenderá los elementos significativos para la competitividad. • Comprenderá la importancia de la productividad. • Comprenderá la importancia de la calidad en los sistemas de servicios. • Identificara las ventajas de cuantificar los costos de calidad. • Comprenderá la clasificación de los costos de calidad. 	

<p>BLOQUE III: CMMMI</p>	
<p>1.1. Introducción 1.2. Representaciones del modelo 1.2.1. Continua 1.2.2. Por Etapas 1.3. Componentes CMMI 1.4. Niveles de Madurez 1.5. Gestión de Requisitos 1.6. Aseguramiento de la calidad 1.7. Beneficios de CMMI</p>	<ul style="list-style-type: none"> • Conocerá las diferentes representaciones del modelo. • Identificará los componentes de CMMI. • Comprenderá los niveles de madurez. • Comprenderá los beneficios del modelo.
<p>BLOQUE IV: MOPROSOFT</p>	
<p>4.1. Arquitectura de Moprosoft 4.2. Propósito 4.3. Descripción del modelo. 4.4. Responsable y autoridad 4.5. Entradas, salidas y productos internos 4.6. Roles y actividades</p>	<ul style="list-style-type: none"> • Conocerá el modelo Moprosoft así como también su origen y propósito. • Comprenderá las diferentes entradas, salidas y productos internos del modelo. • Identificará los Roles del modelo así como también las actividades que se realizan.
<p>BLOQUE V: Desarrollo Iterativo, Evolutivo y Ágil</p>	
<p>5.1. Desarrollo iterativo 5.2. Desarrollo evolutivo y adaptativo 5.3. Desarrollo Ágil 5.3.1. Clasificación de los métodos 5.3.2. Los principios y el manifiesto ágil 5.3.3. Gestión de proyectos ágiles 5.3.4. Enfrentando la comunicación y la retroalimentación 5.3.5. Prácticas simples y herramientas de proyectos 5.3.6. Procesos empíricos vs. Procesos definidos y prescriptivos 5.3.7. Disciplina de sustentabilidad: el roce humano 5.3.8. El equipo como un Sistema Complejo Adaptativo</p>	<ul style="list-style-type: none"> • Conocerá las características de los desarrollos iterativos, evolutivos y ágiles. • Identificará como se clasifican los métodos ágiles. • Conocerá los principios y manifiesto ágil. • Comprenderá las prácticas simples y las herramientas de proyectos. • Identificará las diferencias entre procesos empíricos vs. procesos definidos. • Comprenderá como el equipo se le conoce como un sistema complejo adaptativo.

5. ACTIVIDADES PARA DESARROLLAR LAS COMPETENCIAS

Actividades del maestro.

- Desarrollar actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías que se van aprendiendo en el desarrollo de la asignatura.
- Actividades de inicio: Técnica expositiva, conferencias, lluvia de ideas.
- Actividad de desarrollo: investigación bibliográfica, estudio supervisado, diseño de prácticas.
- Actividad de evaluación: Informe de investigación, Estudio de casos, lista de cotejo.

Actividades del estudiante.

- Actividades de inicio: La pregunta, memoria, lluvia de ideas.
- Actividades de desarrollo: Investigación de tópicos, documentación y debate de resultados, estudio de casos, elaboración de productos de software.
- Actividades finales: Presentación de producto de software.

6. EVALUACIÓN DE LAS COMPETENCIAS

6.1. Evidencias de aprendizaje	6.2. Criterios de desempeño	6.3. Calificación y acreditación
<ul style="list-style-type: none">• La pregunta• Debate• Listas de cotejo• Estudios de casos.• Texto paralelo• Prototipos	<ul style="list-style-type: none">• Obtenga el 80 % de asistencia.• Realiza trabajos de investigación.• Practique una lectura activa, y comprenda los textos de ejemplos a revisar.• Investigue para abundar en fuentes.• Desarrollo de un producto funcional para la aplicación de los conceptos vistos en clase.	<ul style="list-style-type: none">• 10% de asistencia• 50% Calificación en exámenes parciales/Desarrollo de prototipos.• 20 % Exposición de Temas/Estudios de casos de ejemplo.• 10% Tareas.• 10% Texto Paralelo

7. FUENTES DE INFORMACIÓN

Bibliografía:

1. **Ingeniería de software orientada a objetos con UML, Java e Internet, Alfredo Weitzenfeld, Thomson International; Edición: 1.**
2. **Software Engineering: A Practitioner's Approach, Roger Pressman, Bruce R. Maxim, McGraw-Hill Education; Edición: 8th Revised ed.**
3. **Software Engineering, Ian Sommerville, Pearson Educación; 10th ed.**

8. PERFIL DEL PROFESOR:

Egresado de Licenciatura en Informática o Ingeniería en sistemas, egresado de Maestría y experiencia en la industria.