



UASLP
Universidad Autónoma
de San Luis Potosí



**FACULTAD DE
INGENIERÍA**
Área de Ciencias
de la Computación

2304 – INGENIERÍA DE SOFTWARE

Clave de la materia: 2304

Clave Facultad: 2304

Clave U.A.S.L.P.: ----

Clave CACEI: IA

Nivel del Plan de Estudios: I.S.I., I.C.: 3 No. de créditos: 8

Horas/Clase/Semana: 4 Horas totales/Semestre: 64

Horas/Práctica (y/o Laboratorio): 0

Prácticas complementarias: 0

Trabajo extra-clase Horas/Semana: 4

Carrera/Tipo de materia: I.S.I., I.C./Obligatoria

No. de créditos aprobados: ----

Fecha última de Revisión Curricular: 28-abril-2022

Materia y clave de la materia requisito: 2806 – Sistemas

Interactivos (I.S.I.)

OBJETIVO DEL CURSO

Comprender, analizar y aplicar las diferentes metodologías, técnicas y procedimientos de la ingeniería de software, su factibilidad y aplicaciones; asimismo

comprender íntegramente los fundamentos de las metodologías de administración de proyectos y los procesos de desarrollo de software.

CONTENIDO TEMÁTICO

1. FUNDAMENTOS, CONTEXTO E IMPORTANCIA DE LA INGENIERÍA DE SOFTWARE

Tiempo Estimado: 4 hrs.

Objetivo: Conocer los conceptos de software y la Ingeniería de Software, el cuerpo de conocimientos, y su contexto espacio-temporal.

- 1.1. El software y la ingeniería de software (IS)
- 1.2. Cuerpo de conocimiento de la IS
- 1.3. Contexto espacio-temporal de la IS

2. ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE

Tiempo Estimado: 8 hrs.

Objetivo: Conocer y aplicar los fundamentos de la administración de proyectos de software.

- 2.1. Ámbito del problema
- 2.2. Recursos humanos y estimación del esfuerzo
- 2.3. El plan del proyecto y métricas de proyecto

3. PROCESO DE SOFTWARE

Tiempo Estimado: 4 hrs.

Objetivo: Comprender los modelos del ciclo de vida y de los procesos de software así como sus aplicaciones.

- 3.1. Modelos del ciclo de vida y proceso de software
- 3.2. Estado del arte en procesos de software y aplicaciones

4. INGENIERÍA DE REQUERIMIENTOS (IR)

Tiempo estimado: 8 hrs.

Objetivo: Comprender y aplicar los procesos de administración y de ingeniería de requerimientos.

- 4.1. Administración, análisis y especificación de requerimientos
- 4.2. Validación de requerimientos

5. FUNDAMENTOS DE DISEÑO DE SOFTWARE

Tiempo estimado: 30 hrs.

Objetivo: Conocer los métodos y estrategias de diseño, sus procesos y notaciones; así como aplicar el diseño de software a problemas complejos.

- 5.1. Métodos y estrategias de diseño
- 5.2. Procesos y notaciones de diseño
- 5.3. Diseño de Software

6. CONSTRUCCIÓN, EVOLUCIÓN Y EVALUACIÓN DE SOFTWARE

Tiempo estimado: 10 hrs.

Objetivo: Comprender los fundamentos de construcción de software, de la administración de su configuración, su evolución y la evaluación de los atributos del software.

- 6.1. Construcción de software
- 6.2. Administración de la configuración de software
- 6.3. Evolución y evaluación de software

METODOLOGÍA

Exposición del tema por parte del profesor utilizando material audiovisual. Seguir el método de aprendizaje activo, en donde se encargan lecturas a los estudiantes antes de la clase. Fomentar la participación en clase mediante discusiones sobre el tema. Cuando se aplique

se usarán actividades de trabajo colaborativo. Desarrollo de aplicaciones vistas en clase con la guía del profesor, así como ejercicios que el estudiante deberá resolver, siguiendo el método de aprendizaje basado en problemas.

EVALUACIÓN

Se realizarán tres exámenes parciales de forma colegiada en las fechas establecidas por la Facultad, de acuerdo al Reglamento de Exámenes. La calificación de los exámenes parciales estará compuesta en un 80% por

el examen y un 20% por otras actividades (tareas, investigaciones, resolución de problemas, ejercicios, etc.). La calificación del examen ordinario es el promedio de los tres parciales.

BIBLIOGRAFÍA

Bibliografía Básica

Wieggers, K. E., Beatty, J. *Software Requirements*. 3rd Edition, Microsoft Press, 2013.

Sommerville, I. *Software Engineering*. 10th Edition, Pearson, 2015.

Jacobson, I., Booch, G., & Rumbaugh, J. *The Unified Software Development Process*. 2nd Edition, Addison-Wesley, 2004.

Weitzenfeld, A. *Ingeniería de Software Orientada a Objetos con UML, Java e Internet*. Thomson, 2005.

McConnell, S. *Code Complete*. Pearson Education, 2004.

Martin, R. C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Pearson Education, 2017.

Fowler, M., & Beck, K. *Refactoring: Improving the Design of Existing Code*. 2nd Edition, Addison-Wesley Professional, 2018.

Feathers, M. *Working Effectively with Legacy Code*. Prentice Hall Professional, 2004.

Jorgensen, P. *Software Testing: A Craftsman's Approach*. 4th Edition, Auerbach Publications, 2013.

Bibliografía Complementaria

Hunt, A., & Thomas, D. *Pragmatic Thinking and Learning: Refactor Your Wetware*. Addison-Wesley Professional, 2008.

Martin, R. C. *The Clean Coder: A Code of Conduct for Professional Programmers*. Pearson Education, 2011.

Brooks Jr, F. P. *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition, 2/E*. Pearson Education India, 1995.