

**GOBIERNO CONSTITUCIONAL DEL ESTADO LIBRE Y SOBERANO DE OAXACA
INSTITUTO ESTATAL DE EDUCACIÓN PÚBLICA DE OAXACA
COORDINACIÓN GENERAL DE PLANEACIÓN EDUCATIVA
COORDINACIÓN GENERAL DE EDUCACIÓN MEDIA SUPERIOR Y SUPERIOR**

PROGRAMA DE ESTUDIOS

NOMBRE DE LA ASIGNATURA	INGENIERÍA DE SOFTWARE II
-------------------------	---------------------------

CICLO SEXTO SEMESTRE	CLAVE DE LA ASIGNATURA IS-02	TOTAL DE HORAS 80
-------------------------	---------------------------------	----------------------

OBJETIVO(S) GENERAL(ES) DE LA ASIGNATURA

Aplicar los métodos y técnicas existentes para el desarrollo, pruebas, documentación e implantación de un proyecto de software conforme a los requerimientos establecidos en el dominio del proyecto de software. Así como brindar los métodos de ingeniería de software para el mantenimiento y adaptación de los sistemas de información.

TEMAS Y SUBTEMAS

1. IMPLEMENTACIÓN

- 1.1. Selección del entorno de desarrollo
 - 1.1.1. Evaluación y selección de la plataforma y entorno de programación
 - 1.1.2. Evaluación y selección del lenguaje de programación
 - 1.1.3. Evaluación y selección del manejador de bases de datos
- 1.2. Determinación de pautas para los desarrolladores
 - 1.2.1. Asignación de tareas de programación
 - 1.2.2. Políticas de colaboración
 - 1.2.3. Estándares de programación
 - 1.2.4. Consideración de buenas prácticas de programación
- 1.3. Codificación
 - 1.3.1. Elaboración y evaluación de prototipos
 - 1.3.2. Desarrollo e integración de componentes

2. VERIFICACIÓN Y VALIDACIÓN

- 2.1. Definiciones
- 2.2. Diferencia entre defecto-error-falla
- 2.3. Proceso de pruebas
 - 2.3.1. Pruebas de unidad
 - 2.3.2. Pruebas de integración
 - 2.3.3. Pruebas del sistema
 - 2.3.4. Pruebas de aceptación
- 2.4. Técnicas de inspección de software

- 2.5. Técnicas de prueba de software
 - 2.5.1. Pruebas de caja blanca o pruebas estructurales
 - 2.5.2. Pruebas de caja negra o pruebas funcionales
 - 2.5.3. Pruebas de propiedades emergentes del sistema
- 2.6. Generación del plan de pruebas
- 2.7. Diseño de casos y procedimientos de prueba
- 2.8. Ejecución y documentación de casos de pruebas
- 2.9. Análisis de errores y depuración

3. ELABORACIÓN DE MANUALES

- 3.1. Descripción funcional
- 3.2. Manual de instalación
- 3.3. Manual de operación (usuario)
- 3.4. Manual de administrador
- 3.5. Guía de referencia rápida
- 3.6. Manual técnico del sistema
- 3.7. Otros

4. IMPLANTACIÓN

- 4.1. Planeación de la implantación del sistema
 - 4.1.1. Métodos de conversión del sistema
- 4.2. Capacitación de usuarios
 - 4.2.1. Técnicas de capacitación
- 4.3. Retroalimentación
- 4.4. Liberación del sistema
- 4.5. Presentación del producto final

5. MANTENIMIENTO Y ADAPTACIÓN DE SOFTWARE

- 5.1. Dinámica de los cambios en los sistemas
- 5.2. Software legado
- 5.3. Mantenimiento de software
 - 5.3.1. Definición de mantenimiento
 - 5.3.2. Mantenimiento correctivo
 - 5.3.3. Mantenimiento adaptativo
 - 5.3.4. Mantenimiento perfectivo
 - 5.3.5. Mantenimiento preventivo
 - 5.3.6. Proceso de mantenimiento
 - 5.3.7. Proceso de mantenimiento de emergencia
 - 5.3.8. Predicción de mantenimiento
- 5.4. Migración de sistemas
 - 5.4.1. Migración de arquitectura
 - 5.4.2. Migración de interfaz de usuario
 - 5.4.3. Migración de datos

- 5.5. Reingeniería de software
 - 5.5.1. Proceso de reingeniería
 - 5.5.2. Reingeniería de datos
- 5.6. Administración de la configuración
 - 5.6.1. Control de cambios
 - 5.6.2. Control de versiones y liberaciones
- 5.7. Fin de ciclo de vida útil del sistema

ACTIVIDADES DE APRENDIZAJE

Se debe retomar el proyecto con los avances respectivos del análisis y diseño planteados en la materia prerrequisito. El proyecto deberá concluirse al finalizar el curso.

CRITERIOS Y PROCEDIMIENTOS DE EVALUACIÓN Y ACREDITACIÓN

Se realizan tres evaluaciones parciales y una evaluación ordinaria final de la asignatura.

Para las evaluaciones parciales, se deberá realizar un examen escrito y se podrá complementar la evaluación con exámenes prácticos, avances de proyectos, tareas, investigaciones y otras actividades académicas previamente aprobadas de acuerdo con la normatividad Universitaria. Queda a criterio del profesor la ponderación de todas las actividades.

Para la evaluación ordinaria final, se deberá realizar un examen escrito y se podrá complementar la evaluación con proyectos, exposiciones, tareas e investigaciones realizadas a lo largo del semestre. Queda a criterio del profesor la ponderación de todas las actividades.

Para la calificación final de la asignatura, se establece la ponderación de las evaluaciones parciales y ordinaria final con base en la normatividad de la Universidad.

BIBLIOGRAFÍA (TIPO, TÍTULO, AUTOR, EDITORIAL Y AÑO)

Básica:

- Análisis y diseño de aplicaciones informáticas de gestión. Piattini Velhuis, Mario Gerardo; Calvo-Manzano Villalon, José A.; Cervera Bravo, Joaquín; Fernández Sanz, Luis. Ra-Ma. 2003.
- Análisis y diseño orientado a objetos: con UML y el proceso unificado. Schach, Stephen R. McGraw-Hill. 2005, 1ª Edición.
- El lenguaje unificado de modelado, UML 2.0. Booch, Grady; Rumbaugh, James; Jacobson, Ivar. Addison Wesley. 2006, 2ª Edición.
- Ingeniería de software clásica y orientada a objetos. Schach, Stephen R. McGraw-Hill. 2006, 6ª Edición.
- Ingeniería de software explicada. Norris, Mark; Rigby, Peter. Limusa. 1994.
- Ingeniería de software orientada a objetos con UML, Java e internet. Weitzenfeld, Alfredo. Thomson. 2005.
- Ingeniería de software orientado a objetos. Bruegge, Bernd. Pearson Educación. 2002.
- Ingeniería de software: una perspectiva orientada a objetos. Braude, Eric J. Ra-Ma. 2003.
- Ingeniería del software. Pfleeger, Shari Lawrence. Prentice Hall. 2002.
- Ingeniería del software. Sommerville, Ian. Pearson. 2005, 7ª Edición.
- Ingeniería del software: un enfoque práctico. Pressman, Roger S. McGraw-Hill. 2007.
- Software engineering. Sommerville, Ian. Addison Wesley. 2007.
- UML 2. Arlow, Jim; Neustadt, Ila. Anaya Multimedia. 2006.
- UML: El lenguaje unificado de modelado, guía del usuario. Booch, Grady; Rumbaugh, James; Jacobson, Ivar. Addison-Wesley. 2006.

Consulta:

- Análisis estructurado moderno. Yuordon, Edward. Prentice Hall. 1993, 1ª Edición.
- Análisis y diseño de sistemas de información. Senn, James A. McGraw-Hill. 1992, 2ª Edición.
- Análisis y diseño de sistemas. Kendall, Kenneth E.; Kendall, Julie E. Prentice Hall. 2005, 6ª Edición.

- Aprendiendo UML en 24 hrs. Schmuller, Joseph. Pearson. 2000, 1ª Edición.
- Calidad en el desarrollo y mantenimiento del software. Piattini Velthuis, Mario Gerardo; García Rubio, Félix O. Alfaomega. 2003.
- Crystal Reports: resultados profesionales. Peck, George. McGraw-Hill. 2004, 1ª Edición.
- Diseño de interfaces de usuario: estrategias para una interacción persona-computadora efectiva. Shneiderman, Ben. Pearson Addison-Wesley. 2006, 4ª Edición.
- Diseño de sistemas de información: Teoría y práctica. Burch, John G.; Grudnitski, Gary. Limusa. 1999, 5ª Edición.
- El proceso unificado de desarrollo de software. Jacobson, Ivar; Booch, Grady; Rumbaugh, James. Addison Wesley. 2000.
- Elementos de estilo de programación. Kernighan, Brian W.; Plauger, P. J. Diana. 1980, 1ª Edición.
- Head first object-oriented analysis and design. McLaughlin, Brett D.; Pollice Gary; West, Dave. O'Reilly Media. 2004, 1ª Edición.
- Ingeniería de software. Campderrich Falgueras, Benet. UCO. 2003.
- Introducción al proceso software personal. Humphrey, Watts S. Addison-Wesley. 2001, 1ª Edición.
- UML gota a gota. Fowler, Martin; Scott, Kendall. Addison Wesley. 1999.
- UML para programadores Java. Martin, Robert C. Pearson Educación. 2004.
- UML y patrones: introducción al análisis y diseño orientado a objetos y al proceso unificado. Larman, Craig. Prentice Hall. 2003.
- Utilización de UML en ingeniería del software con objetos y componentes. Stevens, Perdita; Pooley, Rob. Addison-Wesley. 2002.
- Utilizar UML en ingeniería de software. Joyanes Aguilar, Luis; Pooley, Rob; Stevens, Perdita. Pearson Educación. 2002.

PERFIL PROFESIONAL DEL DOCENTE

Licenciatura en Informática, Ingeniería en Ciencias Computacionales o afines, con grado de Maestría y preferentemente de Doctorado en Ingeniería de Software o afines. Con experiencia profesional y docente de un año.