

**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	CALIDAD DE SOFTWARE
-------------------------	---------------------

CICLO OCTAVO SEMESTRE	CLAVE DE LA ASIGNATURA IS-04	TOTAL DE HORAS 80
--------------------------	---------------------------------	----------------------

OBJETIVO(S) GENERAL(ES) DE LA ASIGNATURA Comprender y aplicar las métricas, normas y estándares de calidad necesarios en el desarrollo de software.
--

TEMAS Y SUBTEMAS
<ul style="list-style-type: none">1. INTRODUCCIÓN A LA CALIDAD DE SOFTWARE<ul style="list-style-type: none">1.1. Sistemas de calidad1.2. Concepto de calidad de software1.3. Factores de calidad de software1.4. Principios fundamentales del desarrollo de software 2. GESTIÓN DE CALIDAD<ul style="list-style-type: none">2.1. Identificación de clientes2.2. Definición de las necesidades de calidad2.3. Establecer métricas de calidad2.4. Definir estrategia de calidad2.5. Implementar y supervisar programas de calidad 3. MODELOS DE EMPRESA Y PERSONAL PARA DESARROLLO DE SOFTWARE<ul style="list-style-type: none">3.1. Modelo de madurez de la capacidad3.2. Modelo de procesos de software3.3. ISO/IEC 15504 para evaluación de procesos de software3.4. Proceso personal de software3.5. Proceso de equipo de software 4. ESTRATEGIAS DE ANÁLISIS Y DISEÑO<ul style="list-style-type: none">4.1. Análisis<ul style="list-style-type: none">4.1.1. Problemas frecuentes4.1.2. Recomendaciones para análisis

4.2. Diseño

- 4.2.1. Problemas comunes
- 4.2.2. Recomendaciones
- 4.2.3. Principios de diseño
- 4.2.4. Patrones de diseño

5. MÉTRICAS DE SOFTWARE

- 5.1. Métricas de calidad (ISO/IEC 9126)
- 5.2. Propiedades de las métricas
- 5.3. Marcos de validación de métricas
- 5.4. Taxonomía de las métricas

6. HERRAMIENTAS DE CALIDAD DE SOFTWARE

- 6.1. Componentes reutilizables
- 6.2. Jerarquía de herencia
- 6.3. Marco de aplicaciones orientado a objetos
- 6.4. Análisis de dominios
- 6.5. Reestructura y refactorización
- 6.6. Tipos de refactorización

ACTIVIDADES DE APRENDIZAJE

Realizar investigación sobre los temas actuales de las áreas en las unidades de aprendizaje.

Para cada unidad de aprendizaje se diseñarán, desarrollarán o utilizarán programas especializados para que el alumno practique lo expuesto en clases.

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 sistemas de información. Senn, James A. McGraw-Hill. 1992, 2ª Edición.
- Análisis y diseño orientado a objetos con aplicaciones. Booch, Grady. Pearson. 1996, 2ª Edición.
- Calidad de sistemas informáticos. Piattini Velthuis, Mario Gerardo; García Rubio, Félix O.; Caballero, Ismael. Alfaomega Ra-Ma. 2007, 1ª Edición.
- Calidad en el desarrollo y mantenimiento del software. García Rubio, Félix O.; Piattini Velthuis, Mario Gerardo. Alfaomega. 2003.
- Definición y análisis de un proceso de negocios: una guía breve de Six Sigma. Lowenthal, Jeffrey N. Panorama. 2005, 1ª Edición.

- Ingeniería de software: una perspectiva orientada a objetos. Braude, E. J. Alfaomega. 2003.
- Ingeniería del software. Sommerville, Ian. Pearson. 2005, 7ª Edición.
- Ingeniería del software: un enfoque práctico. Pressman, Roger S. McGraw-Hill. 2005, 6ª Edición.
- Introducción al proceso software personal. Humphrey, Watts S. Addison-Wesley. 2001, 1ª Edición.
- La meta: un proceso de mejora continua. Goldratt, Eiyahu M.; Cox, Jeff. Díaz de Santos. 2005, 3ª Edición.
- Patrones de diseño. Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John. Addison-Wesley. 2003, 1ª Edición.
- Six Sigma transaccional para cintas verdes. Windsor, Samuel E. Panorama. 2007, 1ª Edición.

Consulta:

- A framework of software measurement. Zuse, Horst. Walter de Gruyter. 1997.
- A guide to the project management body of knowledge: (pmbok guide). Project Management Institute. PMI. 2008, 4ª Edición.
- Agile processes in software engineering and extreme programming. Succi, Giancarlo; Damián, Ernesto; Scotto, Marco; Concas, Giulio. Springer-Verlag. 2007.
- Análisis y diseño de sistemas. Kendall, Kenneth E.; Kendall, Julie E. Prentice Hall. 2005, 6ª Edición.
- Análisis y diseño orientado a objetos: con UML y el proceso unificado. Schach, Stephen R. McGraw-Hill. 2005, 1ª Edición.
- Aprendiendo UML en 24 hrs. Schmuller, Joseph. Pearson. 2000, 1ª Edición.
- CMMI distilled. Ahern, Dennis M.; Clouse, Aaron; Turner, Richard. Pearson. 2008, 3ª Edición.
- Continuous integration: improving software quality and reducing risk. Duvall, Paul; Matyas, Steve; Glover, Andrew. Addison-Wesley, Signature Series. 2007.
- Effective software project management. Wysocki, Robert K. John Wiley. 2006.
- El lenguaje unificado de modelado, UML 2.0. Booch, Grady; Rumbaugh, James; Jacobson, Ivar. Addison Wesley. 2006, 2ª Edición.
- Estimación de costos y administración de proyectos de software. Jones, Capers. McGraw-Hill. 2008, 2ª Edición.
- Fábricas de software: experiencias, tecnologías y organización. Piattini Velthuis, Mario Gerardo. Alfaomega. 2007, 1ª Edición.
- Head first design patterns. Freeman, Eric; Freeman, Elisabeth. O'Reilly Media. 2004, 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 orientada a objetos con UML, Java e internet. Weitzenfeld, Alfredo. Thomson. 2005.
- Ingeniería de software orientado a objetos. Bruegge, Bernd. Pearson Educación. 2002.
- Medición y estimación del software: técnicas y métodos para mejorar la calidad y la productividad. Piattini Velthuis, Mario Gerardo. Alfaomega. 2008, 1ª Edición.
- Metrics and models in software quality engineering. Kan, Stephen H. Addison Wesley. 2002, 2ª Edición.
- Patterns of enterprise application architecture. Fowler, Martin. Addison-Wesley, Signature Series. 2002.
- Professional software development: shorter schedules, higher quality products, more successful projects, enhanced careers. McConnell, Steve. Addison-Wesley Professional. 2003.
- Refactoring: improving the design of existing code. Fowler, Martin; Beck, Kent; Brant, John; Opdyke, William; Roberts, Don. Addison-Wesley, Pearson Education, Object Technology Series. 1999.
- Six Sigma for managers: 24 lessons to understand and apply six sigma principles in any organization. Brue, Greg. McGraw-Hill. 2005, 1ª Edición.
- SOA design patterns. Erl, Thomas. Prentice Hall. 2009, 1ª Edición.
- Software project survival guide (pro -- best practices). McConnell, Steve. Microsoft Press. 1997, 1ª Edición.
- The capability maturity model: guidelines for improving the software process. Paulk, M.C. Addison Wesley. 1995.
- UML gota a gota. Fowler, Martin; Scott, Kendall. Addison Wesley. 1999.
- UML para programadores Java. Martin, Robert C. Pearson Educación, 2004.
- UML: El lenguaje unificado de modelado, guía del usuario. Booch, Grady; Rumbaugh, James; Jacobson, Ivar. Addison-Wesley. 2006.

- Utilización de UML en ingeniería del software con objetos y componentes. Stevens, Perditá; Pooley, Rob. Addison-Wesley. 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, Administración de TI o afines. Con experiencia profesional y docente de un año.