



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Facultad de Estudios Superiores Aragón
Plan de Estudios



Ingeniería en Computación
Ingeniería de Software

| Clave | Semestre | Créditos | Área | |
|-----------|-------------|----------|---------------------------------------|---------|
| | 6 | 8.0 | Programación e Ingeniería de Software | |
| Modalidad | Curso | | Tipo | Teórico |
| Carácter | Obligatorio | | | |
| Horas | | | | |
| Semana | | | Semestre | |
| Teóricas | 4.0 | | Teóricas | 64.0 |
| Prácticas | 0.0 | | Prácticas | 0.0 |
| Total | 4.0 | | Total | 64.0 |

Seriación indicativa

| | |
|-------------------------------|---|
| Asignatura antecedente | Programación Web 1, Diseño y Análisis de Algoritmos |
| Asignatura subsecuente | Programación Web 2 |

Objetivo general: Aplicar los conocimientos sobre herramientas, fases, procesos y metodologías, para construir productos de software de calidad, basados en estándares de la industria.

Índice temático

| No. | Tema | Horas Semestre | |
|----------------------------|--|----------------|-----------|
| | | Teóricas | Prácticas |
| 1 | INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE | 12.0 | 0.0 |
| 2 | PATRONES DE DISEÑO Y ARQUITECTURA DE SOFTWARE | 14.0 | 0.0 |
| 3 | GESTIÓN DE PROYECTOS Y ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE | 14.0 | 0.0 |
| 4 | SCRUM COMO METODOLOGÍA DE DESARROLLO ÁGIL | 16.0 | 0.0 |
| 5 | LA INGENIERÍA DE SOFTWARE Y EL GOBIERNO DE LAS TECNOLOGÍAS DE LA INFORMACIÓN | 8.0 | 0.0 |
| Total | | 64.0 | 0.0 |
| Suma total de horas | | 64.0 | |



CONSEJO ACADÉMICO DEL ÁREA DE LAS
 CIENCIAS FÍSICO MATEMÁTICAS
 Y DE LAS INGENIERÍAS

Contenido Temático

1. INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE

Objetivo: Conocer los fundamentos de la ingeniería de software, para la inclusión de buenas prácticas en el desarrollo de productos de software de calidad.

- 1.1 Definición de ingeniería de software.
- 1.2 Estructura de la ingeniería de software
 - 1.2.1 Ciclo de vida del desarrollo de software.
 - 1.2.2 Herramientas para el desarrollo de software.
- 1.3 El proceso de desarrollo de software, como instrumento rector en la construcción de software de calidad.
 - 1.3.1 Ejemplos de procesos de desarrollo de software.
 - 1.3.1.1 Unified Process.
 - 1.3.1.2 Personal Software Process.
 - 1.3.1.3 Team Software Process.
- 1.4 Metodologías de desarrollo ágil.
 - 1.4.1 Orígenes de las metodologías de desarrollo ágil.
 - 1.4.2 Metodología Lean.
 - 1.4.3 Xtreme Programming.
 - 1.4.4 SCRUM.

2. PATRONES DE DISEÑO Y ARQUITECTURA DE SOFTWARE

Objetivo: Definir las arquitecturas para el desarrollo de software y aplicar los conocimientos de reutilización de soluciones exitosas a problemas comunes por medio de patrones de diseño.

- 2.1 Origen de los patrones diseño.
- 2.2 Catálogo inicial de patrones.
 - 2.2.1 Tipos de patrones de diseño.
- 2.3 Patrones para la construcción de componentes (Factory Patterns).
 - 2.3.1 Patrón de diseño Factory.
 - 2.3.2 Patrón de diseño Singleton.
- 2.4 Patrones de diseño de estructura.
 - 2.4.1 Patrón Puente (Bridge).
 - 2.4.2 Patrón Adaptador (Adapter).
- 2.5 Patrones de diseño de comportamiento.
 - 2.5.1 Patrón Observer.
 - 2.5.2 Patrón Command.
- 2.6 Patrón Modelo-Vista- Controlador (MVC).

3. GESTIÓN DE PROYECTOS Y ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

Objetivo: Adquirir los conocimientos básicos referentes a la estimación, gestión y seguimiento de un proyecto de desarrollo de software, con herramientas y técnicas formales en el mercado.

- 3.1 Planificación de proyectos de software.
 - 3.1.1 Técnicas de estimación del esfuerzo.
 - 3.1.1.1 Puntos por función.
 - 3.1.1.2 Constructive Cost Model (COCOMO).
- 3.2 Establecimiento de las actividades y asignación de recursos.
 - 3.2.1 Métricas para la gestión de proyectos.
 - 3.2.2 Gráficas de Gantt y ruta crítica.
 - 3.2.3 Seguimiento de las actividades
 - 3.2.4 Gestión de cambios de alcance.
- 3.3 Gestión de riesgos.
 - 3.3.1 Clasificación de riesgos.
 - 3.3.2 Planes de contención y contingencia.
- 3.4 Calidad de software.
 - 3.4.1 Estándares de calidad.
 - 3.4.1.1 ISO 15504. Software Process Improvement and Capability Determination.
 - 3.4.1.2 ISO IEC/25010:2011. Requisitos de Calidad y Evaluación de Sistemas y Software.
 - 3.4.2 Gestión basada en la mejora de procesos.
 - 3.4.2.1 CMMi.
 - 3.4.2.2 Moprosoft.

4. SCRUM COMO METODOLOGÍA DE DESARROLLO ÁGIL

Objetivo: Comprender los conceptos de desarrollo de software mediante la metodología de desarrollo ágil SCRUM, y aplicarlos a un caso práctico.

- 4.1 Orígenes de la metodología.
- 4.2 Roles de la metodología SCRUM.
 - 4.2.1 Scrum Master.
 - 4.2.2 Scrum Team.
 - 4.2.3 Producto Owner.
- 4.3 Estructura de desarrollo basada en ciclos.
 - 4.3.1 Backlog del producto.
 - 4.3.2 Planeación del Sprint.
 - 4.3.3 Ejecución del Sprint.
 - 4.3.4 Revisión y retrospectiva.
- 4.4 Desarrollo de un caso práctico con SCRUM.

5. LA INGENIERÍA DE SOFTWARE Y EL GOBIERNO DE LAS TECNOLOGÍAS DE LA INFORMACIÓN

Objetivo: Comprender cómo se administran las Tecnologías de la Información, a través de los conocimientos de la arquitectura empresarial para el desarrollo de software.

- 5.1 Introducción al gobierno de las tecnologías de la información. IT Governance.
- 5.2 La norma ISO/IEC -38500.
- 5.3 Arquitectura Empresarial.
- 5.4 Dominios del gobierno de TI.
- 5.5 Frameworks para el Gobierno de las T.I.
 - 5.5.1 ITIL.
 - 5.5.2 Cobit 5.
 - 5.5.3 ISO 27000.

| Estrategias didácticas | | Evaluación del aprendizaje | | Recursos | |
|----------------------------------|-----|----------------------------|-----|--------------------------|-----|
| Exposición | (X) | Exámenes parciales | (X) | Aula interactiva | () |
| Trabajo en equipo | (X) | Examen final | (X) | Computadora | (X) |
| Lecturas | () | Trabajos y tareas | (X) | Plataforma tecnológica | (X) |
| Trabajo de investigación | (X) | Presentación de tema | (X) | Proyector o Pantalla LCD | (X) |
| Prácticas (taller o laboratorio) | () | Participación en clase | (X) | Internet | (X) |
| Prácticas de campo | () | Asistencia | () | | |
| Aprendizaje por proyectos | (X) | Rúbricas | () | | |
| Aprendizaje basado en problemas | (X) | Portafolios | () | | |
| Casos de enseñanza | () | Listas de cotejo | () | | |
| Otras (especificar) | | Otras (especificar) | | Otros (especificar) | |

| Perfil profesiográfico | |
|----------------------------|---|
| Título o grado | <ul style="list-style-type: none"> • Poseer un título a nivel licenciatura en Ingeniería en Computación, Ciencias de la Computación, Matemáticas Aplicadas a la Computación o carreras cuyo perfil sea afín al área de Programación e Ingeniería de Software. |
| Experiencia docente | <ul style="list-style-type: none"> • Poseer conocimientos y experiencia profesional relacionados con los contenidos de la asignación a impartir. • Tener la vocación para la docencia y una actitud permanentemente educativa a fin de formar íntegramente al alumno: <ul style="list-style-type: none"> ○ Para aplicar recursos didácticos. ○ Para motivar al alumno. ○ Para evaluar el aprendizaje del alumno, con equidad y objetividad. |
| Otra característica | <ul style="list-style-type: none"> • Poseer conocimientos y experiencia pedagógica referentes al proceso de enseñanza-aprendizaje. • Tener disposición para su formación y actualización, tanto en los conocimientos de su área profesional, como en las pedagógicas. • Identificarse con los objetivos educativos de la institución y hacerlos propios. • Tener disposición para ejercer su función docente con ética profesional: <ul style="list-style-type: none"> ○ Para observar una conducta ejemplar fuera y dentro del aula. ○ Para asistir con puntualidad y constancia a sus cursos. ○ Para cumplir con los programas vigentes de sus asignaturas. |

| Bibliografía básica | Temas para los que se recomienda |
|---|----------------------------------|
| Geoff, H. (2014). <i>Governance of Enterprise IT Based on COBIT 5: A Management Guide.</i> UK:-It Governance Ltd. | 5 |
| Gido, J. y Clements, J. (2014). <i>Administración exitosa de proyectos.</i> Australia: CENGAGE Learning. | 3 |
| Green, D. (2016). <i>Scrum: novice to ninja.</i> USA: SitePoint Pty. | 4 |
| Pressman, R. S. (2010). <i>Ingeniería de Software, un enfoque práctico.</i> México: Mc. Graw Hill. | 1, 2, 3, 4 y 5 |

| | |
|--|----------------|
| Rivera, M. F. (2015). <i>Administración de proyectos: guía para el aprendizaje.</i> México: Pearson Educación de México. | 3 |
| Selig, J. D. (2008). <i>Implementing Effective It Governance and It Management.</i> USA: Van Haren Pub. | 5 |
| Sommerville, I. (2012). <i>Ingeniería de Software.</i> México: Pearson. | 1, 2, 3, 4 y 5 |
| Toro, L. F. (2013). <i>Administración de proyectos de informática.</i> Bogotá: Ecoe Ediciones. | 3 |

| Bibliografía complementaria | Temas para los que se recomienda |
|--|---|
| Debrauwer, L. (2013). <i>Patrones de diseño en Java. Los 23 Modelos De Diseño. Descripción y Solución Ilustradas en UML 2 y Java.</i> España: Eni-Ediciones. | 2 |
| Dominik, M. (2015). <i>The Scrum Culture: Introducing Agile Methods in Organizations.</i> Alemania: Springer. | 4 |
| Gamma, E. (1997). <i>Design patterns: elements of reusable object-oriented software.</i> United States of America: Addison-Wesley. | 2 |
| Mendes, E. (2014). <i>Practitioner's Knowledge Representation A Pathway to Improve Software Effort Estimation.</i> New York: Springer-Verlag New York Inc. | 3 |
| Velthuis, P. (2008). <i>Medición y estimación del software. Técnicas y métodos para mejorar la calidad y la productividad.</i> México: Alfaomega. | 3 |