



**PLAN DE ESTUDIOS (PE):** Licenciatura en Ingeniería en Tecnologías de la

Información

**ÁREA:** Área de Modelado de Sistemas

**ASIGNATURA:** Control de Calidad de Software

**CÓDIGO:** ITIS 261

**CRÉDITOS:** 6

**FECHA:** 01 de octubre de 2018





**1. DATOS GENERALES**

<b>Nivel Educativo:</b>	<u>Licenciatura</u>
<b>Nombre del Plan de Estudios:</b>	<u>Licenciatura en Ingeniería en Tecnologías de la Información</u>
<b>Modalidad Académica:</b>	<u>Presencial</u>
<b>Nombre de la Asignatura:</b>	<u>Control de Calidad de Software</u>
<b>Ubicación:</b>	<u>Nivel Formativo</u>
<b>Correlación:</b>	
<b>Asignaturas Precedentes:</b>	<u>Ingeniería de Software II</u>
<b>Asignaturas Consecuentes:</b>	<u>Ninguna</u>

**2. CARGA HORARIA DEL ESTUDIANTE**

Concepto	Horas por semana		Total de horas por periodo	Total de créditos por periodo
	Teoría	Práctica		
<b>Horas teoría y práctica (16 horas = 1 crédito)</b>	3	2	90	6





### 3. REVISIONES Y ACTUALIZACIONES

Autores:	Juan Manuel González Calleros Josefina Guerrero García Martínez Mirón Erika Anabel María Luz Adolfina Sánchez Gálvez Mario Anzures García
Fecha de creación:	1 de octubre de 2018
Fecha de la última actualización:	10 de octubre de 2018
Fecha de aprobación por parte de la academia de área, departamento u otro.	
Sinopsis de la revisión y/o actualización:	Curso de nueva creación.

### 4. PERFIL DESEABLE DEL PROFESOR (A) PARA IMPARTIR LA ASIGNATURA:

Disciplina profesional:	Tecnologías de la Información, Sistemas Computacionales, Ciencias de la Computación, Ingeniería de Software
Nivel académico:	Maestría
Experiencia docente:	Mínima de 2 años
Experiencia profesional:	Mínima de 2 años

**5. PROPÓSITO:** Desarrollar software, a partir del uso de estrategias y técnicas que sirven para transmitir un cierto nivel de certeza sobre la calidad de un sistema de información.

### 6. COMPETENCIAS PROFESIONALES:

Competencias específicas de ITI:

- Diseña y desarrolla sistemas innovadores empleando la vanguardia de la tecnología y metodologías de las TI para apoyar el desarrollo productivo en los ámbitos de la administración pública y privada, las redes sociales y de conocimiento.
- Aplica la capacidad crítica, de análisis y síntesis para integrar el pensamiento creativo, crítico y procesos cognitivos en el desarrollo de proyectos de software innovadores en TI; asumiendo una actitud proactiva, congruente con sus conocimientos, habilidades y valores al interior de grupos multidisciplinarios de trabajo.

Justificación:

El desarrollo de software de calidad es un proceso que requiere de la adopción de estrategias formales apoyadas de herramientas de software modernas. Esto se logra con la adopción de metodologías para la evaluación y validación de un producto de software.





**7. CONTENIDOS TEMÁTICOS**

<b>Unidad de Aprendizaje</b>	<b>Contenido Temático</b>	<b>Referencias</b>
1. Control de calidad del software	1.1 Concepto de calidad de software 1.2 Aseguramiento de Calidad 1.3 Metodologías y modelos de madurez en el desarrollo de software de calidad 1.4 Procesos de desarrollo de software y mejora de procesos 1.5 Proceso de revisión y Programa de inspección 1.6 Métricas del producto 1.7 Análisis de componentes de software. 1.8 Desarrollo ágil y su relación con la calidad global 1.9 Refactorización	Walkinshaw, N. (2017). <i>Software Quality Assurance: Consistency in the Face of Complexity and Change</i> . Springer.  Sacha, Krzysztof (Editor) (2006). <i>Software engineering techniques: Design for quality</i> , Springer-Verlag, Germany.  Braude, Eric J., Bernstein, Michael E. (2016), <i>Software engineering: Modern approaches</i> , Second Edition, Waveland Press, USA.
2. Fundamentos de las pruebas de software	2.1 ¿Por qué es necesario probar? 2.2 ¿Qué es la prueba (testing)? 2.3 Principios generales de las pruebas 2.4 La psicología de las pruebas 2.5 Los modelos de desarrollo de software 2.6 Los niveles de prueba o fases 2.7 Los tipos u objetivos de prueba 2.8 Pruebas de mantenimiento	Walkinshaw, N. (2017). <i>Software Quality Assurance: Consistency in the Face of Complexity and Change</i> . Springer.  Ghazali, Umer W. (2014) <i>Software testing: Essential skills for first time testers</i> , Amazon Digital Services LLC, USA.  Watkins, John (2009). <i>Agile testing: How to succeed in a extreme testing environment</i> , Cambridge University Press, USA.
3. Técnicas de diseño de prueba	3.1 Identificar y diseñar casos de prueba 3.2 Categorías de técnicas de diseño de prueba 3.3 Basada en especificación (caja negra) 3.4 Basada en estructura (caja blanca)	Walkinshaw, N. (2017). <i>Software Quality Assurance: Consistency in the Face of Complexity and Change</i> . Springer.  Myers, Glenford J., Sandler, Corey, Badgett, Tom (2011). <i>The art of software testing</i> , 3rd Edition, Wiley, USA.





Unidad de Aprendizaje	Contenido Temático	Referencias
	3.5 Técnicas basadas en la experiencia 3.6 Elección de las técnicas de prueba 3.7 Técnicas estáticas 3.7.1 Proceso de prueba 3.7.2 Uso de herramientas	Watkins, John (2009). Agile testing: How to succeed in a extreme testing environment, Cambridge University Press, USA.
4. Gestión de pruebas	4.1 Organización de pruebas 4.2 Planificación y estimación de prueba 4.3 El seguimiento y control de los progresos de prueba 4.4 Gestión de la configuración 4.5 Riesgo y pruebas 4.6 Gestión de incidentes	Walkinshaw, N. (2017). Software Quality Assurance: Consistency in the Face of Complexity and Change. Springer.  Lewis, William E. (2008). Software testing and continuous quality improvement, Third Edition, Auerbach Publications, USA.  Hunt, Andy, Thomas, Dave (2003) Pragmatic unit testing, The Pragmatic Bookshelf, USA.
5. Herramientas de soporte de pruebas	5.1 Tipos de herramientas de prueba 5.2 El uso efectivo de herramientas, beneficios y riesgos potenciales 5.3 Uso de una herramienta 5.3.1 Pruebas de Unidad 5.3.2 Integración Continua 5.3.3 DevOPs	Walkinshaw, N. (2017). Software Quality Assurance: Consistency in the Face of Complexity and Change. Springer.  Kaczanowski, Tomek (2013). Practical unit testing with Junit and Mockito, Tomasz Kaczanowski.  Pajankar, Ashwin (2017). Python unit test automation: Practical techniques for Python developers and testers, 1st Edition, Apres, USA.  Gundecha, Unmesh (2012) Selenium testing tools cookbook, Packt Publishing, USA.





### 8. ESTRATEGIAS, TÉCNICAS Y RECURSOS DIDÁCTICOS

Estrategias y técnicas didácticas	Recursos didácticos
<p>Estrategias de aprendizaje:</p> <ul style="list-style-type: none"> <li>• Lectura y comprensión,</li> <li>• Reflexión,</li> <li>• Comparación,</li> <li>• Resumen.</li> </ul> <p>Estrategias de enseñanza:</p> <ul style="list-style-type: none"> <li>• ABP,</li> <li>• Aprendizaje activo,</li> <li>• Aprendizaje cooperativo,</li> <li>• Aprendizaje colaborativo,</li> <li>• Basado en el descubrimiento.</li> </ul> <p>Ambientes de aprendizaje:</p> <ul style="list-style-type: none"> <li>• Aula,</li> <li>• Simuladores.</li> </ul> <p>Actividades y experiencias de aprendizaje:</p> <ul style="list-style-type: none"> <li>• Visita a empresas.</li> </ul>	<ul style="list-style-type: none"> <li>• Proyector,es,</li> <li>• Plumón y pizarrón,</li> <li>• Software multimedia</li> <li>• Herramientas de modelado</li> <li>• Equipos de cómputo</li> <li>• Pizarrón inteligente</li> <li>• Aplicaciones colaborativas.</li> <li>• Libros y revistas.</li> </ul>

### 9. EJES TRANSVERSALES

Eje (s) transversales	Contribución con la asignatura
Formación Humana y Social	Promover el trabajo en equipo en el desarrollo de Software de calidad.
Desarrollo de Habilidades en el uso de las Tecnologías de la Información y la Comunicación	Promover el uso de herramientas tecnológicas que fomenten el desarrollo de Software de Calidad.
Desarrollo de Habilidades del Pensamiento Complejo	Contribuir al proceso de representación de un problema mediante el planteamiento de escenarios de prueba de Software.
Lengua Extranjera	Lectura de bibliografía especializada en el área de calidad de software.
Innovación y Talento Universitario	Representación de la solución de diversos problemas del mundo real basados en la construcción de software de calidad.

### 10. CRITERIOS DE EVALUACIÓN





Criterios	Porcentaje
• Participación en clase	10 %
• Tareas	20%
• Exposiciones	20%
• Prácticas de laboratorio	20%
• Proyecto final	30%
Total	100%
	100%

### 11. REQUISITOS DE ACREDITACIÓN

Estar inscrito como alumno en la Unidad Académica en la BUAP
Asistir como mínimo al 80% de las sesiones
La calificación mínima para considerar un curso acreditado será de 6.
No se debe contar con antecedentes comprobados de copia o plagio de prácticas o proyectos durante el curso.
Cumplir con las actividades académicas y cargas de estudio asignadas que señale el PE

#### Notas:

- a) La entrega del programa de asignatura con sus respectivas actas de aprobación, deberá realizarse en formato electrónico, vía oficio emitido por la Dirección o Secretaría Académica a la Dirección General de Educación Superior.
- b) La planeación didáctica deberá ser entregada a la coordinación de la licenciatura en los tiempos y formas acordados por la Unidad Académica.

