

## 1. Datos Generales de la asignatura

<b>Nombre de la asignatura:</b>	VERIFICACIÓN Y VALIDACIÓN DEL SOFTWARE.
<b>Clave de la asignatura:</b>	TDF-1803
<b>SATCA1:</b>	3-2-5
<b>Carrera:</b>	Ingeniería en Sistemas Computacionales

## 2. Presentación

### Caracterización de la asignatura

Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales la formación de personal con un dominio sobre el proceso de verificación y validación con un enfoque que promueve la calidad del software.

La importancia de esta asignatura se centra en la realización de pruebas completas al software antes de ser liberado hacia el usuario final, esto debido a que se ha comprendido la importancia de las pruebas de calidad y su impacto en la industria del software.

Los aspectos principales a considerar son las revisiones y las pruebas como parte del ciclo de vida del software que se utilizan para detectar fallas en las diferentes etapas del mismo. Se centra en la detección de defectos en el producto, dándole mucha importancia a las revisiones.

La verificación y la validación del software incluyen un conjunto de procedimientos, actividades, técnicas y herramientas, que se utilizan paralelamente al desarrollo del mismo para asegurar que el producto cubre los diferentes aspectos de calidad. El objetivo es prevenir las fallas desde los requerimientos hasta su implementación.

### Intención didáctica

La asignatura es teórico – práctica y permite desarrollar en el estudiante la habilidad para la aplicación de las diferentes técnicas de verificación y validación de software, considerando los principios de la ingeniería de software, para lo cual el temario se organiza en cinco bloques.

En el bloque uno, introducción a la verificación y validación del software, se retoma de manera importante la calidad del software como fundamento para conocer y comprender el proceso de verificación y validación del software, así como la identificación de los estándares de la IEEE que le son aplicados.

El bloque dos apoyará en la definición de las revisiones al software, así como el proceso de pruebas de software. En el bloque tres se detalla el proceso de verificación del software.

En el bloque cuatro se hace una relación entre el modelado y el proceso de verificación. En el bloque cinco, se aborda el proceso de implementación y las diferentes herramientas que se pueden utilizar en el proceso.

En el último bloque, se plantea el proceso a desarrollar en el proceso de validación del software por parte del cliente.

### 3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico Superior de Irapuato Fecha: 15 al 17 de Abril de 2013. El Instituto Tecnológico Superior de Acayucan revisó en el mes de Noviembre de 2017.	Academia de Ingeniería en Sistemas Computacionales – Desarrollo de Software Academia de Ingeniería en Sistemas Computacionales.	Análisis y adecuación por competencias del módulo de la especialidad "Ingeniería de Software" de la carrera de Ingeniería en Sistemas Computacionales.

### 4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
<p>Aplicar técnicas para verificaciones y validaciones al software, considerando los aspectos de revisión y pruebas como parte del ciclo de vida para detectar fallas en el desarrollo del mismo.</p> <p><b>Competencias instrumentales:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis.</li> <li>• Capacidad de organizar y planificar.</li> <li>• Conocimientos básicos de la carrera.</li> <li>• Comunicación oral y escrita.</li> <li>• Habilidades del manejo de la computadora.</li> <li>• Habilidad para buscar y analizar información proveniente de fuentes diversas.</li> <li>• Solución de problemas.</li> <li>• Toma de decisiones.</li> </ul> <p><b>Competencias interpersonales:</b></p> <ul style="list-style-type: none"> <li>• Capacidad crítica y autocrítica.</li> <li>• Trabajo en equipo.</li> <li>• Habilidades interpersonales.</li> </ul> <p><b>Competencias sistémicas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica.</li> <li>• Habilidades de investigación.</li> <li>• Capacidad de aprender.</li> <li>• Capacidad de generar nuevas ideas (creatividad).</li> </ul>

- Búsqueda del logro.

### 5. Competencias previas

- Identifica el proceso de ingeniería de software
- Conoce el ciclo de vida
- Utiliza herramientas de modelado
- Capacidad de análisis
- Capacidad de crítica constructiva

### 6. Temario

No.	Temas	Subtemas
1	Introducción al proceso de verificación y validación.	1.1 Contextualización de la verificación y validación. 1.2 Terminología del proceso. 1.3 El proceso de la verificación y validación. 1.4 Tipos generales de los errores. 1.5 Responsabilidad de pruebas. 1.6 Organigrama de proceso de testing (un modelo propuesto). 1.7 Costos del error.
2	Estudios de volúmenes y velocidad	2.1. Tipos de pruebas. 2.2 Cobertura de las pruebas. 2.3 Preparación de la prueba. 2.4 Productos de la prueba. 2.5 Criterios para la realización de pruebas. 2.6 Plan Pruebas (validación y verificación). 2.7 Estructura de los casos de Prueba. 2.8 Conceptos Generales los diseños de las pruebas (validación y verificación). 2.9 Reporte y Seguimiento de errores. 2.10 Informe de la Prueba. 2.11 Fuentes de información de QA para el control estadística o métricas. 2.12 Control estadístico vs métricas. 2.13 Importancia de la calidad, las métricas y el control estadístico
3	Verificación	3.1 Marco de Referencia para el desarrollo de software. 3.2 Herramientas para apoyar al proceso y la ejecución de las revisiones de software.

		<p>3.3 Manejo de Requerimientos (Verificación). 3.4 Verificación en este proceso. 3.5 Entradas propuestas para el proceso de verificación de requerimientos. 3.6 Método de verificación. 3.7 Aspectos a verificar en esta etapa. 3.8 Entendimiento de problema (Verificación). 3.9 Revisión general de requerimientos.</p>
4	Modelado.	<p>4.1 Modelado de pruebas con UML. 4.2 Cumplimiento de la especificación en los requerimientos. 4.3 Importancia en la efectividad en el diseño. 4.4 Patrones (tipos de patrones, como utilizar los patrones para validar). 4.5 Contratación y Outsourcing.</p>
5	Implementación.	<p>5.1. Implementación. 5.2 Entradas para pruebas. 5.3 Plan de pruebas (estrategia de prueba, ambientes, test team, atacar y asegurar regresión). 5.4 Ejecución de tipos generales de pruebas. 5.5 Caja negra y caja blanca. 5.6 Otros tipos de test. 5.7 GUI, Funcionalidad, Performance, entre otros. 5.7.1 Documentación (técnica y de usuario). 5.7.2 Seguridad. 5.7.3 Diseño de las pruebas.</p>

## 7. Actividades de aprendizaje de los temas

Introducción al proceso de la verificación y validación.	
Competencias	Actividades de aprendizaje
<p>Específica(s): Identifica el proceso de verificación y validación del software.</p> <p>Genéricas: Habilidad para buscar y analizar información, capacidad para trabajar en equipo, aplicar los conocimientos en la práctica.</p>	<ul style="list-style-type: none"> <li>Investigar los conceptos de calidad del software.</li> <li>Analiza el proceso de verificación y validación del software.</li> <li>Desarrollar un glosario de términos</li> <li>Integración de la verificación y validación al proceso de desarrollo de software.</li> <li>Conocer el papel de los estándares en el proceso de verificación y validación del software.</li> </ul>
Pruebas.	

Competencias	Actividades de aprendizaje
<p>Específica(s): Identifica las características y los tipos de pruebas que se le aplican al software.</p> <p>Genéricas: Habilidad para buscar y analizar información, capacidad para trabajar en equipo, aplicar los conocimientos en la práctica.</p>	<ul style="list-style-type: none"> <li>Investigar la definición y características de las revisiones de software.</li> <li>Diseñar un caso de prueba</li> <li>Realizar un reporte de pruebas y seguimiento de errores.</li> <li>Utilizar formatos propuestos para la documentación de las pruebas de software.</li> </ul>
Verificación.	
Competencias	Actividades de aprendizaje
<p>Específica(s): Analiza el proceso de verificación al software.</p> <p>Genéricas: Capacidad para buscar y analizar información, capacidad para trabajar en equipo, aplicar los conocimientos en la práctica.</p>	<ul style="list-style-type: none"> <li>Investigar el proceso de la verificación.</li> <li>Realizar en un caso práctico el manejo de requerimientos por medio de un método de verificación.</li> </ul>
Modelado.	
Competencias	Actividades de aprendizaje
<p>Específica(s): Utiliza herramientas para el modelado en el proceso de verificación y validación del software.</p> <p>Genéricas: Capacidad para buscar y analizar información, capacidad para trabajar en equipo, aplicar los conocimientos en la práctica</p>	<ul style="list-style-type: none"> <li>Investigar el uso de herramientas para apoyar al proceso y la ejecución de las revisiones de software.</li> <li>Investigar el uso de herramientas para apoyar al proceso, el diseño y la ejecución de las pruebas de software.</li> <li>Modelar casos prácticos de pruebas.</li> </ul>
Implementación.	
Competencias	Actividades de aprendizaje
<p>Específica(s): Planea el proceso de verificación y validación del software, y aplicar métricas para evaluar los resultados finales.</p> <p>Genéricas: Habilidad para buscar y analizar información, capacidad para trabajar en equipo, aplicar los conocimientos en la</p>	<ul style="list-style-type: none"> <li>Investiga el proceso de planeación.</li> <li>Ejecuta pruebas y analiza los resultados</li> </ul>

práctica.	
-----------	--

Validación y logística de pruebas	
Competencias	Actividades de aprendizaje
<p>Específica(s): Analizar el proceso de verificación y validación del software, y aplicar métricas para evaluar los resultados finales.</p> <p>Genéricas: Habilidad para buscar y analizar información, capacidad para trabajar en equipo, aplicar los conocimientos en la práctica.</p>	<ul style="list-style-type: none"> <li>• Analiza e interpreta los resultados de las pruebas.</li> <li>• Elabora carta de aceptación del cliente.</li> <li>• Documenta el seguimiento a cambios del proyecto</li> </ul>

### 8. Práctica(s)

<ol style="list-style-type: none"> <li>1. Seleccionar un caso práctico de desarrollo de software para someterlo al proceso de verificación y validación del software.</li> <li>2. Utilizar las herramientas para apoyar al proceso de verificación y validación del software.</li> <li>3. Diseñar un plan de verificación y validación del software, así como aplicar las métricas para evaluar los resultados arrojados por la verificación y validación del software.</li> <li>4. Diseñar un formato de pruebas.</li> <li>5. Realizar visitas a empresas dedicadas al desarrollo de software.</li> <li>6. Desarrollar una mesa de debates, en la cual se cuente con la presencia de un experto en desarrollo, a fin de conocer las experiencias con respecto a los procesos de verificación y validación del software.</li> </ol>
---

### 9. Proyecto de asignatura

<p>El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:</p> <ul style="list-style-type: none"> <li>• <b>Fundamentación:</b> marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.</li> <li>• <b>Planeación:</b> con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.</li> <li>• <b>Ejecución:</b> consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que</li> </ul>
---

implica el desempeño de las competencias genéricas y específicas a desarrollar.

- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

Además de considerar la revisión del diseño de la arquitectura, es los siguientes aspectos:

- Mapas conceptuales, cuestionario sobre conceptos relacionados con la arquitectura.
- Revisión de la documentación del modelado.

## 10. Evaluación por competencias

La evaluación debe ser continua y formativa por lo que se debe considerar el desempeño de cada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Aplicación de un examen de diagnóstico al inicio del curso.
- Ponderar tareas.
- Información obtenida durante las investigaciones solicitadas, plasmadas en documentos escritos o digitales
- Dar seguimiento al desempeño en el desarrollo del temario (dominio de los conceptos, capacidad de la aplicación de los conocimientos en problemas reales y de ingeniería)
- Se recomienda utilizar varias técnicas de evaluación con un criterio específico para cada una de ellas (teórico-práctico).
- Examen escrito correspondiente a cada unidad.
- Ejercicios dentro de laboratorios.
- Examen práctico (resolviendo aspectos reales) correspondiente a cada unidad.
- Desarrollo de un proyecto final que integre todas las unidades de aprendizaje
- Presentación del proyecto final. (Informe, presentación y defensa congruencia del proyecto final).
- Uso de una plataforma educativa en internet la cual puede utilizarse como apoyo para crear el portafolio de evidencias del alumno (integrando: tareas, prácticas, evaluaciones, etc.)

## 11. Fuentes de información

1. Test Process Improvement, Tim Koomen and Martin Pol. Addison-Wesley. 1999
2. Software Testing and Continuous Quality Improvement, William E. Lewis. Aurebach. 2000
3. Effective Methods for Software Testing (second edition), William E. Perry. Wiley Computer Publishing. 2000
4. Software Verification and Validation, Steven R. Rakitin. Artech House. 1997
5. UML y Patrones Craig Larman. Pearson. 1999
6. Software Testing and Quality Assurance Theory and Practice, Kshirasagar Naik & Priyadarshi Tripathy, Wiley.
7. Verification and Validation in Systems Engineering, Assessing UML/SysML Design Models, Mourad

Debbabi, Fawzi Hassaine - Yosr Jarraya, Springer.