Datos Generales de la asignatura

Nombre de la asignatura:	**Verificación y Validación de Software**
Clave de la asignatura:	**ISC-2605**
SATCA[1]:	**2-2-4**
Carrera:	**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 la asignatura se organiza en cinco unidades. En la unidad uno, introducción a la verificación y validación del software, se retoma de manera importante la calidad del software como fundamento para conocer y comprender

[1] Sistema de Asignación y Transferencia de Créditos Académicos

el proceso de verificación y validación del software, así como la identificación de los estándares de la IEEE que le son aplicados.

La unidad dos apoyará en la definición de las revisiones al software, así como el proceso de pruebas de software.

En la unidad tres se detallan el proceso de verificación del software.

En la unidad cuatro de hace una relación entre el modelado y el proceso de verificación.

En la unidad cinco, se aborda el proceso de implementación y las diferentes herramientas que se pueden utilizar en el proceso.

En la última unidad, 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
Dirección General de Institutos Descentralizados, México D.F. Fecha: 12 al 14 de septiembre de 2012.	Representantes: Tecnológico de Estudios Superiores de Coacalco. Instituto Tecnológico Superior de Atlixco, Comalcalco, Fresnillo, Santiago Papasquiaro, Tepexi de Rodríguez, Zapopan.	Análisis y adecuación por competencias del módulo de la especialidad “Ingeniería de Software" de la carrera de Ingeniería en Sistemas Computacionales.
Tecnológico de Estudios Superiores de Coacalco Fecha: 10 al 14 de agosto de 2015.	Representantes: Tecnológico de Estudios Superiores de Coacalco.	Actualización del módulo de la especialidad “Ingeniería de Software" de la carrera de Ingeniería en Sistemas Computacionales
Tecnológico de Estudios Superiores de Coacalco Fecha: 21 al 22 de junio de 2019.	Representantes: Tecnológico de Estudios Superiores de Coacalco.	Actualización de asignaturas del módulo de la especialidad "Ingeniería de Software" de la carrera de Ingeniería en Sistemas Computacionales. Análisis, formulación de competencias, actualización de temario,

Tecnológico de Estudios Superiores de Coacalco Octubre del 2025.	Representantes: Docentes de la Academia de Ingeniería en Sistemas Computacionales del Tecnológico de Estudios Superiores de Coacalco.	formulación de actividades y prácticas de la especialidad de "Ingeniería de Software y Arquitectura de Sistemas" de la carrera de Ingeniería en Sistemas Computacionales.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Maneja 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 mismo.

5. Competencias previas

- Identifica el proceso de ingeniería de software
- Conoce el ciclo de vida de un software
- 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, defectos y fallos. 1.5 Responsabilidad de pruebas. 1.6 Organigrama de proceso de testing (un modelo propuesto). 1.7 Costos del error
2	Pruebas.	2.1 Tipos de pruebas. 2.2 Cobertura de las pruebas. 2.3 Preparación de la prueba. 2.3.1 Formatos de matrices de casos de pruebas con Organigrama de proceso de testing (un modelo propuesto). 2.3.2 Costos que tengan casos de pruebas de funcionalidad positiva,

		negativa, excepciones y no afectación 2.4 Productos de la prueba. 2.5 Criterios para la realización de pruebas. 2.5.1 Niveles de Pruebas. 2.6 Plan Pruebas (validación y verificación). 2.7 Estructura de los casos de Prueba de acuerdo con el Software a validar (Aplicativos web, procesos batch, ETL, SP, bases de datos, aplicativos locales) 2.8 Conceptos Generales los diseños de las pruebas (validación y verificación). 2.9 Reporte y Seguimiento de errores, defectos y fallos. 2.10 Informe de la Prueba de acuerdo con los 7 principios de las pruebas. 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. 2.13.1 JIRA para reportes de defectos y seguimiento de ciclos de pruebas 2.13.2 MAGNIFAI pseudocódigo para automatización de pruebas 2.1.13.3 JAVA Y SELENIUM para automatización de pruebas 2.13.4CUCUMBER automatización de pruebas
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. 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. 3.10 Fase de manejo de requerimientos.
4	Modelado	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.
5	Implementación.	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.
6	Validación y logística de pruebas.	6.1 Pruebas y aceptación del cliente 6.2 Entrega de proceso de pruebas. 6.3 Formalización y cierre del proyecto. 6.4 Monitoreo y seguimiento del proyecto. 6.5 Formalización de cambios. 6.6 Administración de defectos.

7. Actividades de aprendizaje de los temas

UNIDAD 1. Introducción al proceso de la verificación y validación	
Competencias	Actividades de aprendizaje
Específica(s): Identifica el proceso de verificación y validación del software Genéricas: • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones.	• Investiga los conceptos de calidad del software. • Analiza el proceso de verificación y validación del software. • Desarrolla un glosario de términos • Integra de la verificación y validación al proceso de desarrollo de software. • Conoce el papel de los estándares en el proceso de verificación y validación del software.

• Conocimiento de una segunda lengua	
UNIDAD 2. Pruebas	
Competencias	Actividades de aprendizaje
Específica(s): Identifica las características y los tipos de pruebas que se le aplican al software Genéricas: • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Conocimiento de una segunda lengua.	• Investiga la definición y características de las revisiones de software. • Diseña un caso de prueba • Realiza un reporte de pruebas y seguimiento de errores. • Utiliza formatos propuestos para la documentación de las pruebas de software
UNIDAD 3. Verificación	
Competencias	Actividades de aprendizaje
Específica(s): Analiza el proceso de verificación al Software Genéricas: • Capacidad crítica y autocrítica • Trabajo en equipo • Habilidades interpersonales • Capacidad de comunicarse con profesionales de otras áreas • Compromiso ético	• Investiga el proceso de la verificación. • Realiza en un caso práctico el manejo de requerimientos por medio de un método de verificación
UNIDAD 4. Modelado	
Competencias	Actividades de aprendizaje
Específica(s):	• Investiga el uso de herramientas para apoyar al proceso y la

Utiliza herramientas para el modelado en el proceso de verificación y validación del software Genéricas: • Capacidad crítica y autocrítica. • Trabajo en equipo • Habilidades interpersonales • Capacidad de comunicarse con profesionales de otras áreas • Compromiso ético	ejecución de las revisiones de software. • Investiga el uso de herramientas para apoyar al proceso, el diseño y la ejecución de las pruebas de software. • Modela casos prácticos de pruebas
UNIDAD 5. Implementación	
Competencias	Actividades de aprendizaje
Específica(s): Planea el proceso de verificación y validación del software, y aplicar métricas para evaluar los resultados finales Genéricas: • Capacidad de aplicar los Conocimientos en la práctica • Habilidades de investigación • Capacidad de aprender • Capacidad de generar nuevas ideas (Creatividad) • Capacidad de diseñar y gestionar proyectos • Preocupación por la calidad	• Investiga el proceso de planeación. • Ejecuta pruebas y analiza los resultados
UNIDAD 6. Validación y logística de pruebas	
Competencias	Actividades de aprendizaje
Específica(s): Analiza el proceso de verificación y validación del software, y aplicar métricas para evaluar los resultados finales Genéricas: • Capacidad de aplicar los conocimientos en la práctica	• Analiza e interpreta los resultados de las pruebas. • Elabora carta de aceptación del cliente. • Documenta el seguimiento a cambios del proyecto.

• Habilidades de investigación • Capacidad de generar nuevas ideas (Creatividad) • Capacidad de diseñar y gestionar proyectos • Preocupación por la calidad	

8. Práctica(s)

- Seleccionar un caso práctico de desarrollo de software para someterlo al proceso de verificación y validación del software.
- Utilizar las herramientas para apoyar al proceso de verificación y validación del software
- 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.
- Diseñar un formato de pruebas.
- Realizar visitas a empresas dedicadas al desarrollo de software.
- 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.

9. Proyecto de asignatura

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:

- **Fundamentación:** 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.

- **Planeación:** 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.

- **Ejecución:** 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 implica el desempeño de las competencias genéricas y especificas 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.

10. Evaluación por competencias

Se recomienda evaluar los siguientes puntos:

- Evaluación Teórica.
- Prácticas de laboratorio.
- Desarrollo y evaluación de un proyecto de software.

La ponderación de la evaluación debe ser propuesta por la academia correspondiente de cada institución.

11. Fuentes de información

- Cecil Martín, R. (2004). UML para Programadores en Java. Addison Wesley.
- Lewis, W. (2016). Software Testing and Continuous Quality Improvement. Aurebach.
- Perry, W. (2007). Effective Methods for Software Testing (third edition). WileyComputer Publishing.
- Fisher, M. (2007). Software Verification and Validation and Engineering and Scientific Approach. Springer.
- Booch, G., Rumbaugh, J. y Jacobson, I. (2007). El lenguaje unificado de modelado Manual de referencia. Pearson Education.
- Naik, K. and Tripathy, P. (2011). Software Testing and Quality Assurance Theory and Practice. Wiley.
- Debbabi, M., Hassaine, F. and Jarraya, Y. (2010). Verification and Validation in Systems Engineering, Assessing UML/SysML Design Models. Springer.