



1. Datos Generales de la asignatura

Nombre de la asignatura:	Verificación y Validación de Software
Clave de la asignatura:	ISC-2105
SATCA¹:	2-2-4
Carrera:	Ingeniería en Sistemas Computacionales

2. Presentación

Caracterización de la asignatura

La asignatura de Verificación y Validación se encuentra en el IX semestre y forma parte de la especialidad Ingeniería de software, junto con el grupo de asignaturas: Programación de Móviles; Ingeniería de Requerimientos; Arquitectura y Diseño de Software; Modelo de Desarrollo Integral (CMMI) y Proceso Personal para el Desarrollo de Software.

El desarrollo de la competencia de la asignatura por parte del estudiante aporta al perfil del Ingeniero en Sistemas Computacionales, en el desarrollo y administración de software para apoyar la productividad y competitividad de las organizaciones cumpliendo con estándares de calidad, así como, 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 para 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 de este. 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

El enfoque por competencias requiere que las actividades prácticas promuevan el desarrollo de habilidades para la investigación y tecnológicas por equipo. Durante la implementación de las actividades se propician procesos intelectuales como inducción-deducción y análisis-síntesis con la intención de generar una actividad intelectual compleja, es conveniente que el facilitador acompañe a sus alumnos para que ellos realicen las evaluaciones formativas y mejores sus productos de aprendizaje.

La asignatura es teórica-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 seis 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 la norma ISO 29119, que le son aplicados. El bloque dos apoyará en la

¹ Sistema de Asignación y Transferencia de Créditos Académicos

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 seis, se plantea a desarrollar el proceso de validación del software que incluya Pruebas y aceptación del cliente, entrega de proceso de pruebas, formalización y cierre del proyecto. El papel del docente es plantear la aplicación del proceso de validación del software a un producto de software, que el alumno ha desarrollado en semestres anteriores.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Tecnológico Nacional de México, Campus Comalcalco Tabasco, julio 2021.	Tecnológico Nacional de México, Campus Comalcalco Tabasco	Reunión de Academia de Sistemas Computacionales, según consta minuta con fecha 13 de julio de 2021.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Aplica las 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

Competencias específicas:

- Desarrolla soluciones de software, considerando la metodología y herramientas para la elaboración de un proyecto aplicativo en diferentes escenarios.
- Aplica metodologías e instrumentos, para garantizar la gestión adecuada de un proyecto de software.
- Desarrolla e implementa aplicaciones con tecnología móvil, a partir de los conceptos, técnicas y herramientas.
- Elabora un plan de mejora de procesos y aprende la metodología para implementar dicha mejora.

Competencias genéricas:

- Utilizar herramientas de modelado de datos.
- Capacidad de análisis.
- Capacidad de crítica constructiva
- Habilidades de gestión de información (habilidad para buscar y analizar información proveniente de fuentes diversas).
- Capacidad de resolución de problemas.
- Trabajo en equipo.

- Habilidades en el uso de las tecnologías de la información y de la comunicación.

6. Temario

Temas		Subtemas
No.	Nombre	
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.	Pruebas.	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 de 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. 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).

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>
6.	Validación y logística de pruebas.	<p>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.</p>

7. Actividades de aprendizaje de los temas

1. Introducción al proceso de verificación y validación.	
Competencias	Actividades de aprendizaje
<p>Específica:</p> <ul style="list-style-type: none"> Identifica el proceso de verificación y validación del software. <p>Genéricas:</p> <ul style="list-style-type: none"> Capacidad de análisis y síntesis. Capacidad de organizar y planificar. Comunicación oral y escrita. Solución de problemas. Toma de decisiones. Habilidad para buscar y analizar información proveniente de fuentes diversas. 	<ul style="list-style-type: none"> Comunicar y presentar la asignatura: encuadre, empleo de recursos educativos abiertos (REA), entrega de productos y evidencias de aprendizaje en la plataforma, netiqueta y tiempo de acompañamiento docente. Organizar al grupo en equipos para Investigar analiza el proceso de verificación y validación del software a través de un cuadro sinóptico. Se tiene que subir en formato pdf para su evaluación y retroalimentación. Desarrolla un glosario de términos y subirlo a la plataforma en el foro electrónico. Realizar en equipos una práctica para: Aplicar la Integración de la verificación y validación al proceso de desarrollo de software. Desarrolla cuadro sinóptico para conocer el papel de los estándares en el proceso de verificación y validación del software. Reflexiona sobre los requerimientos de desarrollo de software y participa en el foro. Retroalimentar los temas tratados en plenaria por parte del facilitador y los compañeros.
2. Pruebas.	
Competencias	Actividades de aprendizaje
<p>Específica:</p> <p>Desarrolla un plan de prueba de</p>	<ul style="list-style-type: none"> Organizar al grupo para investigar la definición y características de las revisiones

<p>software. Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Solución de problemas. • Toma de decisiones. • Habilidad para buscar y analizar información proveniente de fuentes diversas 	<p>de software para determinar el plan de prueba.</p> <ul style="list-style-type: none"> • Organizar al grupo en equipos para el diseño de un caso de prueba de software. Realizar un reporte de pruebas y seguimiento de errores. • Utiliza formatos propuestos para la documentación de las pruebas de software. • Retroalimentar los temas tratados en plenaria por parte del facilitador y los compañeros.
3. Verificación	
Competencias	Actividades de aprendizaje
<p>Específica(s): Realiza el manejo de requerimientos por medio de un método de verificación.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Solución de problemas. • Toma de decisiones. • Habilidad para buscar y analizar información proveniente de fuentes diversas 	<ul style="list-style-type: none"> • Organizar al grupo para investigar el proceso de verificación. Subir el reporte de lectura de investigación en formato pdf para su valoración y retroalimentación en la plataforma. • Realiza una práctica relacionada con un caso práctico en el manejo de requerimientos por medio de un método de verificación. Emplear rúbrica de informe de prácticas. • Retroalimentar los temas tratados en plenaria por parte del facilitador y los compañeros.
4. Modelado.	
Competencias	Actividades de aprendizaje
<p>Específica(s): Desarrolla un plan de prueba para un caso de uso planteado.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Solución de problemas. • Toma de decisiones. • Habilidad para buscar y analizar información proveniente de fuentes 	<ul style="list-style-type: none"> • Organizar al grupo en equipos para continuar con el proyecto en la fase de diseño de un plan de prueba para un caso en UML. Revisar avances y retroalimentar a los equipos. • Organizar al grupo en equipos para investigar el uso de herramientas para apoyar al proceso, el diseño y la ejecución de las pruebas de software. Subir el reporte de investigación para su evaluación y retroalimentación. • Realizar la práctica en equipos de aplicar Patrones (tipos de patrones) al diseño del plan de prueba. • Retroalimentar los temas tratados en plenaria por parte del facilitador y los compañeros.

diversas.	
5. Implementación	
Competencias	Actividades de aprendizaje
<p><i>Específica:</i> Ejecuta las estrategias de pruebas (caja blanca, de caja negra y los tipos de tests) en la implementación de las pruebas diseñadas en el plan de prueba.</p> <p><i>Genéricas:</i></p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Solución de problemas. • Toma de decisiones. • Habilidad para buscar y analizar información proveniente de fuentes diversas. 	<ul style="list-style-type: none"> • Organizar al grupo en equipo para Investigar el proceso de planeación. Subir el reporte de investigación para su valoración y retroalimentación. • Evaluar la siguiente fase del proyecto: Ejecuta pruebas y analiza los resultados. • Realiza un reporte de pruebas y seguimiento de errores • Retroalimentar los temas tratados en plenaria por parte del facilitador y los compañeros.
6. Validación y logística de pruebas.	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Interpreta el proceso de verificación validación del software, y aplicar métricas para evaluar los resultados finales.</p> <p><i>Genéricas:</i></p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Solución de problemas. • Toma de decisiones. • Habilidad para buscar y analizar información proveniente de fuentes diversas. 	<ul style="list-style-type: none"> • Organiza al grupo en equipos para presentar avances del proyecto para analizar e interpretar los resultados de las pruebas. • Elabora carta de aceptación del cliente. • Documenta el seguimiento a cambios del proyecto. • Integrar todas las fases del proyecto para su evaluación final. • Retroalimentación de todos los temas de la asignatura en plenaria. Evaluación sumativa.

8. Práctica(s)

<ol style="list-style-type: none"> 1. Seleccionar un caso práctico de desarrollo de software para someterlo al proceso de verificación y validación del software. 2. Utilizar las herramientas para apoyar al proceso de verificación y validación del software. 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.

4. Diseñar un formato de pruebas.
5. Realizar visitas a empresas dedicadas al desarrollo de software.
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.

9. Proyecto de asignatura

Proyecto: Diseño de un plan de prueba para un caso en UML de desarrollo de software para someterlo al proceso de verificación y validación del software.

Objetivo: Desarrollar un plan de prueba para un caso UML de desarrollo de software para someterlo al proceso de verificación y validación del software.

- **Fundamentación:** El plan de pruebas de software se elabora para atender los objetivos de calidad en un desarrollo de sistemas, encargándose de definir aspectos como por ejemplo los módulos o funcionalidades sujeto de verificación, tipos de pruebas, entornos, recursos asignados, entre otros aspectos. Para su desarrollo se requiere el desarrollo de las habilidades genéricas como son liderazgo, herramientas de investigación y el uso de la tecnología para verificar y validar software.
- **Planeación:** Se recomienda un plan de pruebas de software, considerando los siguientes pasos:
 - 1.- Analizar los requerimientos de desarrollo de software
 - 2.- Identificar las funcionalidades nuevas a probar
 - 3.- Identificar las funcionalidades de sistemas existentes que deben probarse
 - 4.- Definir la estrategia de pruebas
 - 5.- Definir los criterios de inicio, aceptación y suspensión de pruebas
 - 6.- Identificar los entornos (ambientes) requeridos.
 - 7.- Determinar necesidades de personal y entrenamiento
 - 8.- Establecer la metodología y procedimientos de prueba
- **Ejecución:** consiste en el desarrollo plan de pruebas de software, realizada por parte de los estudiantes con asesoría del docente, se inicia con la solicitud de Prueba", que se da inicio al proceso de pruebas definido. Es responsabilidad del líder de proyecto elaborar esta solicitud facilitando toda la documentación que se requiere, los requisitos asociados, los ejecutables de la aplicación o módulo, los manuales de instalación, configuración y usuario.
- Continúa con "Crear Plan de Prueba"- El cual tiene el objetivo fundamental de establecer la planificación en general del proceso de pruebas en el laboratorio de calidad. En este momento se registran los datos iniciales del plan, se asigna el miembro del equipo que asumirá el rol de Especialista de Pruebas.

Por último, ejecución del plan de prueba el responsable de su ejecución en este caso es el probador, o los probadores que integren el equipo de pruebas. La primera

actividad que se debe realizar es la definición de las opciones de ejecución del caso de prueba, que establecen cuál será el entorno en el que se ejecuten y las configuraciones. Del segundo juego de datos en adelante se reproduce la grabación y se evalúan todas las combinaciones restantes. Si se detecta algún error durante la realización de las comprobaciones se debe entonces generar una No Conformidad (NC) y son estas precisamente el principal artefacto que se obtiene de este subproceso.

- **Evaluación:**

El plan de prueba de desempeño debe aplicar al código fuente de software, considerando los siguientes elementos.

- Métodos.
- Atributos.
- Variables. Conexión a la base de datos.
- Componentes.
- Excepciones.

Las pruebas unitarias se deben aplicar:

- Diferentes escenarios de pruebas.
- Criterios de aceptación.
- Resultados de las pruebas.

10. Evaluación por competencias

Evaluación formativa:

Producto o Evidencia de aprendizaje	Instrumento de evaluación
Cuadro sinóptico	<p>Categorías del cuadro sinóptico de la lista de verificación.</p> <ul style="list-style-type: none"> – El cuadro sinóptico contiene tema o idea central, subtemas o ideas principales, secundarias e información esencial. – El cuadro sinóptico está ordenado, sin faltas de ortografía y con creatividad. – El contenido es claro y está bien distribuido. – Los subtemas están representados llaves. – No hay conceptos repetidos.
Mesa de debates	<p>Rúbrica de presentación de temas relacionados verificación y validación del software, considerando las categorías siguientes:</p> <ul style="list-style-type: none"> – Expongo y argumento mis ideas perfectamente, valorando y respetando las intervenciones y



	<p>opiniones de mis compañeros y compañeras.</p> <ul style="list-style-type: none"> - Utilizo sin dificultad las tecnologías de la información y la comunicación para obtener información. - Asumo mi rol sin interferir en el trabajo de los demás y apporto ideas al grupo.
<p>Caso práctico de desarrollo de software para someterlo al proceso de verificación y validación del software</p>	<p>Lista de cotejo de reporte de practica:</p> <ul style="list-style-type: none"> - Presenta los elementos indispensables: nombre del equipo, institución, nombre de la práctica, datos generales, nombre del curso, nombre del profesor y fecha. - Se evalúan críticamente todas las mediciones empleadas en la verificación del software. - Se evalúan críticamente todas las mediciones empleadas en la validación del software. - Se observa el procedimiento realizado en cada práctica. - Se subió el reporte de lectura a la plataforma para su evaluación.
<p>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</p>	<p>El plan de verificación y validación, que se evaluará a través de lista de cotejo, el cual debe contener:</p> <ol style="list-style-type: none"> 1.- Analizar los requerimientos de desarrollo de software 2.- Identificar las funcionalidades nuevas a probar 3.- Identificar las funcionalidades de sistemas existentes que deben probarse 4.- Definir la estrategia de pruebas 5.- Definir los criterios de inicio, aceptación y suspensión de pruebas 6.- Identificar los entornos (ambientes) requeridos. 7.- Determinar necesidades de personal y entrenamiento. 8.- Establecer la metodología y procedimientos de prueba.
<p>Evaluación sumativa:</p> <ol style="list-style-type: none"> 1. Participación en clases 2. Evidencias de productos y desempeños en la plataforma 3. Portafolio de aprendizaje personal 4. Evaluación del proyecto 	



11. Fuentes de información

Referencias:

- Bill Laboon. (2016). *A Friendly Introduction to Software Testing*. NJ, Estados Unidos: CreateSpace Independent Publishing Platform.
- Blanco-Llano, J., & Rodríguez-Hernández, A. (2011). Revisión, Verificación Y Validación en Un Proceso De Desarrollo De Software. *Ingeniería Industrial*, 32(1), 28–36.
- Blanquicett, L. A., Bonfante, M. C., & Acosta-Solano, J. (2018). Prácticas de Pruebas desde la Industria de Software. La Plataforma ASISTO como Caso de Estudio. *Información Tecnológica*, 29(1), 11–18. <https://doi.org/10.4067/S0718-07642018000100011>
- Brian, H. (2015). *Software Testing: An Istqb-BCS Certified Tester Foundation Guide*. 3rd Ed. Illinois Estados Unidos. BSC.
- Kenett Ron S, Ruggeri Fabrizio, F. y Frederick W. (2018). *Analytic Methods in Systems and Software Testing*. Blacksburg, VA, Estados Unidos. Wiley ed.
- Lewis, William E. (2009). *Software testing and continuous quality improvement*. United States of America: Gunasekaran Veerapillai. Technical Contributor.
- Mera Paz, J. A. (2016). Análisis Del Proceso De Pruebas De Calidad De Software. *Ingeniería Solidaria*, 12(20), 163–176. <https://doi.org/10.16925/in.v12i20.1482>
- Ramos, K. (2020). *Estrategias de Localización de Selenium WebDriver: Guía Práctica para Potenciar las Pruebas Automatizadas*. España: Edición Kindle.
- Rex, B. (2016). *Advanced Software Testing Vol. 1*. 2nd Edition: Guide to the ISTQB
- Rex, B. (2016). *Advanced Software Testing Vol. 2*. 2nd Edition: Guide to the ISTQB
- Rex, B. (2016). *Advanced Software Testing Vol. 3*. 2nd Edition: Guide to the ISTQB

Recursos educativos abiertos:

- Bayona-Oré, S., Chamilco, J., & Perez, D. (2019). Mejora de Procesos Software: Gestión de Requisitos, Verificación y Validación. (Spanish). CISTI (Iberian Conference on Information Systems & Technologies / Conferência Ibérica de Sistemas e Tecnologias de Informação) Proceedings, 1–5.
- DesarrolloWeb.com. 25 may 2017. Desarrollo de pruebas del software. [Archivo de Vídeo]. Youtube. https://www.youtube.com/watch?v=qkeD94Kj_UQ
- Appxolot. (15/05/2020). CURSO GRATIS sobre Introducción a las Pruebas de Software - Testing-Parte 1/3. Youtube. <https://www.youtube.com/watch?v=ib4WlbOmD4s>
- Quality-Stream. 23 feb 2021. Cómo Crear un Plan Pruebas de Software | Norma ISO 29119. [Archivo de Vídeo]. Youtube. <https://www.youtube.com/watch?v=0anZpU5W0Z8>