

1. DATOS DE LA ASIGNATURA

Nombre de la asignatura:	Pruebas de Software
Carrera:	Ingeniería en Sistemas Computacionales
Clave de la asignatura:	ISC-2102
SATCA	2-2-4

2. HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Culiacán. Culiacán Sinaloa México. 30 de abril de 2021.	<ul style="list-style-type: none">• Norma Rebeca Godoy Castro• María del Rosario González Álvarez• Mirna del Rosario Quevedo Camacho	Elaboración de asignatura: Pruebas de Software.

3. PRESENTACIÓN

Caracterización de la asignatura:

La materia Pruebas de Software forma parte de la especialidad de Ingeniería de Software de la carrera de Ingeniería en Sistemas Computacionales. Se relaciona previamente con la asignatura de Fundamentos de Ingeniería de Software. Se recomienda su impartición en el séptimo semestre.

Debido a que el Software en nuestros días juega un rol sumamente importante tanto social como económico, los profesionales del software están presionados a enfocarse en la calidad. Conceptos como estabilidad, escalabilidad, eficiencia, facilidad de uso, confiabilidad y seguridad se relacionan a la calidad de un producto bien desarrollado. En la cadena de valor del desarrollo de un software específico, el proceso de prueba es clave a la hora de detectar errores o fallas.

El proceso de prueba es un proceso técnico especializado de investigación que requiere de profesionales altamente capacitados en lenguajes de desarrollo, métodos, técnicas de pruebas y

herramientas especializadas.

Esta materia aporta al perfil del egresado las competencias necesarias para desarrollar el proceso de pruebas utilizando diferentes técnicas y niveles de prueba con el propósito de coadyuvar en la calidad de un producto software. Además, promueve la aplicación de estos conocimientos en el ciclo completo de las pruebas que deberá emplearse en las distintas fases y pasos del proceso de desarrollo del software.

Intención didáctica

Esta materia debe ser abordada bajo un enfoque teórico-práctico en el cual el maestro fungirá como un facilitador del desarrollo de las competencias generales, específicas y genéricas en el alumno. El alumno deberá participar en cada una de las estrategias de aprendizaje diseñadas por el maestro, donde éstas favorecerán principalmente el saber y el saber hacer.

La materia se organiza en cuatro unidades que pretenden acercar al alumno con las pruebas de software como un proceso planeado, documentado y debidamente gestionado; aplicable a un software para determinar su calidad. La intención didáctica de cada unidad es la siguiente:

En la Unidad 1 se abordan conceptos fundamentales relativos a las pruebas del software, así como reconocer la importancia que tienen éstas para lograr la calidad de un producto software.

En la unidad 2 se abordan métodos técnicos para diseñar casos de prueba efectivos a partir de diferentes coberturas. Estas coberturas permiten medir qué tanto se examinará un software en base a un determinado criterio. Así como, la aplicación de pruebas automatizadas.

En la Unidad 3 se aborda la aplicación y gestión del proceso de diseño de Pruebas Dirigido por Modelos. Se discuten las fases del proceso de pruebas tales como análisis y especificación de requerimientos, diseño de software, diseño intermedio, diseño detallado, implementación, integración, liberación del sistema, operación y mantenimiento, así como la implementación del proceso de pruebas.

En la Unidad 4 se aborda el conocimiento de la metodología de pruebas ágiles, sus principios y estrategias, así como la distinción de los cuatro cuadrantes del Desarrollo Guiado por Pruebas (TDD).

Es importante que el estudiante comprenda y valore las actividades realizadas en el curso, debido al impacto que tiene la calidad del software. Por lo que el alumno deberá desarrollar su capacidad de aplicar los conocimientos en la práctica, de organizar y planificar, de resolver problemas, de trabajar en equipos interdisciplinarios, de diseñar y gestionar proyectos y de liderazgo bajo un compromiso ético.

4. COMPETENCIAS A DESARROLLAR:

Competencia general:

Diseñar el proceso de pruebas orientado a asegurar y comprobar la calidad del software.

Competencias específicas:

- Reconocer el impacto de las pruebas para lograr la calidad del software.
- Implementar métodos técnicos para diseñar casos de prueba efectivos a partir de diferentes coberturas, así como la automatización.
- Implementar el proceso de diseño de Pruebas Dirigido por Modelos.
- Conocer la metodología de pruebas ágiles, principios y estrategias apoyándonos en los cuatro cuadrantes del TDD.

Competencias genéricas:

Competencias instrumentales:

- Capacidad de organizar y planificar.
- Habilidad para buscar y analizar información de diversas fuentes.
- Solución de problemas.

Competencias interpersonales:

- Trabajo en equipo.
- Compromiso ético dentro y fuera del aula.

Competencias sistémicas:

- Capacidad de aplicar los conocimientos en la práctica.

5. COMPETENCIAS PREVIAS

- Identificar la historia del software, así como las características de los paradigmas estructurado y orientado a objetos.
- Realizar cada una de las etapas del ciclo de vida de desarrollo de software, utilizando modelos, técnicas y herramientas.

6. TEMARIO

Unidad	Temas	Subtemas
1	Fundamentos de las pruebas	1.1 Por qué probar el software 1.2 Cuando el software va mal. 1.3 Objetivos de las pruebas de software. 1.4 Origen de los defectos. 1.4.1 Defectos en los requerimientos y la especificación. 1.4.2 Defectos en el diseño. 1.4.3 Defectos en el código. 1.4.4 Defectos en las pruebas. 1.5 Diseño de pruebas basado en modelos.

		<p>1.5.1 Fundamentos de pruebas de software.</p> <p>1.5.2 Actividades de pruebas de software</p> <p>1.5.3 Niveles de pruebas basadas en actividades del software</p> <p>1.5.4 Criterios de cobertura.</p> <p>1.5.5 Diseño de pruebas basado en modelos.</p> <p>Capítulo 3 (Burnstein, 2002) Capítulo 1 y 2 (Paul, 2017)</p>
2	Criterios de cobertura de las pruebas	<p>2.1 Particionando espacios de entrada</p> <p>2.2 Cobertura de Grafos</p> <p>2.3 Cobertura lógica</p> <p>2.4 Automatización de pruebas</p> <p>2.4.1 Casos de prueba</p> <p>2.4.2 Herramientas para automatización</p> <p>Capítulo 3, 6, 7 y 8 (Paúl, 2017)</p>
3	Proceso de pruebas	<p>3.1 Gestión del proceso de pruebas</p> <p>Introducción</p> <p>3.1.2 Análisis y especificación de requerimientos</p> <p>3.1.3 Diseño de software</p> <p>3.1.4 Diseño intermedio</p> <p>3.1.5. Diseño detallado</p> <p>3.1.6 Implementación</p> <p>3.1.7 Integración</p> <p>3.1.8 Liberación del sistema</p> <p>3.1.9 Operación y mantenimiento</p> <p>3.1.10 Implementando el proceso de pruebas</p> <p>3.2 Plan de pruebas</p> <p>Capítulo 10 y capítulo 11 (Pául, 2017)</p>
4	Pruebas ágiles	<p>4.1 ¿Qué son las pruebas ágiles?</p> <p>4.2 Principios de pruebas ágiles</p> <p>4.3 Propósito de las pruebas ágiles</p> <p>4.4 Cuadrantes de pruebas ágiles</p> <p>4.4.1 Cuadrante 1. Desarrollo guiado por pruebas (TDD)</p> <p>4.4.2 Cuadrante 2. Pruebas enfocadas al negocio</p> <p>4.4.3 Cuadrante 3. Pruebas integrales enfocadas al cliente.</p> <p>4.4.4 Cuadrante 4. Pruebas enfocadas a la calidad</p> <p>Capítulo 1,2, 6 y 12 (Crispin & Gregory, 2009)</p>

7. DESARROLLO DE COMPETENCIAS ESPECÍFICAS

Competencias específicas	Actividades de aprendizaje
Unidad 1. Fundamentos de las pruebas. <ul style="list-style-type: none"> Reconocer el impacto de las pruebas para lograr la calidad del software. 	<ul style="list-style-type: none"> Investigar los errores de software más costosos y realizar plenaria. Elaborar un mapa mental/ conceptual del origen de los defectos en el software. Analizar artículos mediante debates o mesas redondas.
Unidad 2. Criterios de cobertura de las pruebas. <ul style="list-style-type: none"> Implementar métodos técnicos para diseñar casos de prueba efectivos a partir de diferentes coberturas, así como la automatización. 	<ul style="list-style-type: none"> Realizar ejercicios de pruebas y su procedimiento para un proyecto de software. Investigar en equipos el uso de herramientas para la automatización de las pruebas y ponerlo en el grupo.
Unidad 3. Proceso de pruebas <ul style="list-style-type: none"> Implementar el proceso de diseño de Pruebas Dirigido por Modelos. 	<ul style="list-style-type: none"> Realizar tabla comparativa de los diseños y compartirlas en el grupo. Elaborar mapa mental/ conceptual del proceso de pruebas. Realizar el proceso de pruebas a partir de un caso de estudio.
Unidad 4. Pruebas ágiles <ul style="list-style-type: none"> Conocer la metodología de pruebas ágiles, principios y estrategias apoyándonos en los cuatro cuadrantes del TDD. 	<ul style="list-style-type: none"> Realizar un glosario de términos acerca de la metodología ágil. Realizar una investigación en las diferentes fuentes bibliográficas de casos de éxito de pruebas ágiles. Desarrollar un caso práctico utilizando los cuatro cuadrantes del TDD.

8. SUGERENCIAS DIDÁCTICAS PARA EL DESARROLLO DE COMPETENCIAS GENÉRICAS.

- Desarrollar actividades que motiven al estudiante a la investigación documental sistemática y organizada.
- Propiciar el uso de las nuevas tecnologías en el desarrollo de los contenidos de la asignatura.
- Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración de los estudiantes.
- Desarrollar actividades de aprendizaje que propicien la aplicación de los conceptos que se van aprendiendo en el desarrollo de la asignatura.
- Propiciar el uso adecuado de conceptos y de terminología científico-tecnológica relacionada

con la asignatura.

- Proponer problemas reales que permitan al estudiante la integración de contenidos de la asignatura y entre distintas asignaturas, para su análisis y solución.

9. SUGERENCIAS DE EVALUACIÓN

Diagnóstica	Formativa	Sumativa
<ul style="list-style-type: none">• Realización de plenaria donde los alumnos expresen sus debilidades y fortalezas con respecto al tema de pruebas del software, así como los conocimientos previos del proceso de desarrollo de software.	<ul style="list-style-type: none">• Realización de actividades lúdicas para la valoración sobre el conocimiento y aplicación de los principios, prácticas y valores del proceso de pruebas.	<ul style="list-style-type: none">• Exámenes parciales (50%)• Desempeño en actividades lúdicas (20%)• Proyecto (30%)

10. FUENTES DE INFORMACIÓN

- **Libro de texto**
 - Ammann, Paul; Offutt, Jeff. *Introduction to Software Testing*. Ed. Cambridge University Press, segunda edición 2017.
 - Crispin, Lisa; Gregory, Janet. *Agile Testing A practical guide for Testers and Agil Team*. Ed Pearson, Education Inc, 2009.
- **Lecturas complementarias**
 - Burnstein, I. *Practical Software Testing: a process-oriented approach*. Ed. Springer. ISBN 0-387-95131-8 2003.
 - Rainsberger, J.B. *JUnit Recipes: Practical Methods for Programmer Testing*. Ed. Manning Publications 2004.

11. PERFIL DEL PROFESOR QUE IMPARTIRÁ LA MATERIA

- Ingeniero en Sistemas Computacionales, Licenciado en Informática o carrera afín con experiencia práctica en aplicación del Proceso de Pruebas de Software.
- Maestro en Ciencias Computacionales.
- Doctor en Ciencias Computacionales.