

1. DATOS GENERALES DE LA ASIGNATURA

Nombre de la asignatura:	Fundamentos de testing
Clave de la asignatura:	TIF-2303
SATCA¹:	3-2-5
Carrera:	Ingeniería Informática

2. PRESENTACIÓN

Caracterización de la asignatura
<p>La asignatura cubrirá los siguientes temas:</p> <ul style="list-style-type: none"> ● Introducción al testing: conceptos fundamentales, objetivos y beneficios del testing. ● Tipos de pruebas: pruebas unitarias, pruebas de integración, pruebas de aceptación, pruebas de rendimiento, etc. ● Herramientas de testing: herramientas populares como Jira y JUnit, etc. ● Planificación y ejecución de pruebas: cómo planificar y realizar pruebas para asegurar la calidad de una aplicación. <p>La asignatura se desarrollará a través de una combinación de clases magistrales, ejercicios prácticos, proyectos en grupo y discusiones en línea. Se invitará a especialistas en testing para que demuestren cómo utilizar herramientas populares y compartan su experiencia en el campo.</p>
Intención didáctica
<p>Proporcionar a los estudiantes una comprensión sólida de los diferentes tipos de pruebas y las herramientas y técnicas necesarias para probar una aplicación. A través de una combinación de clases magistrales, ejercicios prácticos, proyectos en grupo y discusiones en línea, los estudiantes aprenderán cómo planificar y realizar pruebas para asegurar la calidad de una aplicación.</p> <p>Para lograr esto, se utilizarán metodologías activas de aprendizaje, como la discusión en grupo y el trabajo en equipo, para fomentar la participación activa de los estudiantes y promover el pensamiento crítico. Se invitará a especialistas en testing para que demuestren cómo utilizar herramientas populares y compartan su experiencia en el campo, proporcionando a los estudiantes una visión del mundo real del testing.</p> <p>La evaluación se basará en la participación en clase, el desempeño en ejercicios prácticos y proyectos en grupo, y un examen final escrito, para asegurar que los estudiantes han adquirido las habilidades y técnicas adecuadas.</p>

¹ Sistema de Asignación y Transferencia de Créditos Académicos

3. Participantes en la actualización, el diseño, consolidación y/o seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Apatzingán, Michoacán. A 03 de febrero de 2023.	Academia de Informática	Reunión de Academia 2023-01

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Proporcionar a los estudiantes los conocimientos y habilidades necesarios para comprender los diferentes tipos de pruebas y diseñar un plan de pruebas eficaz para un proyecto de software.

5. Competencias previas

<p>Las competencias previas necesarias para la asignatura Fundamentos de Testing incluyen:</p> <ul style="list-style-type: none"> ● Conocimientos básicos de programación: Los estudiantes deben tener un conocimiento básico de un lenguaje de programación para poder entender cómo se escriben y ejecutan pruebas automatizadas. ● Conocimientos básicos de sistemas: Los estudiantes deben tener un conocimiento básico de los sistemas operativos y las plataformas de desarrollo de software. ● Conocimiento de metodologías de desarrollo de software: Los estudiantes deben tener una comprensión básica de las metodologías de desarrollo de software, como ágiles o cascada, para entender cómo se planifican y realizan las pruebas en un entorno de desarrollo. ● Conocimientos básicos de metodologías ágiles: Es recomendable que los estudiantes tengan un conocimiento básico de las metodologías ágiles, ya que se discutirán en la asignatura.
--

6. Temario

No.	Temas	Subtemas
1	Fundamentos de testing	<ol style="list-style-type: none"> 1. Principios y conceptos básicos 2. Objetivo del testing 3. Niveles de pruebas

		4. Tipos de pruebas
2	Técnicas estáticas	<ol style="list-style-type: none"> 1. Técnicas estáticas y el proceso de pruebas 2. Proceso de revisiones 3. Análisis estático con herramientas
3	Técnicas de diseño de pruebas	<ol style="list-style-type: none"> 1. Proceso de desarrollo de prueba 2. Categorías de las técnicas de diseño de prueba 3. Técnicas basadas en la especificación o de caja negra 4. Técnicas basadas en la estructura o de caja blanca 5. Técnicas basadas en la experiencia
4	Plan de pruebas	<ol style="list-style-type: none"> 1. Importancia del plan de pruebas 2. Diseño y planificación del plan de pruebas 3. Ejecución y monitoreo del plan de pruebas 4. Documentación y reporte los resultados de las pruebas
5	Herramientas	<ol style="list-style-type: none"> 1. Herramientas para plan de pruebas 2. Herramientas de testing

7. Actividades de aprendizaje de los temas

1. Introducción al testing	
Competencias	Actividades de aprendizaje
<p>Específica:</p> <ul style="list-style-type: none"> ● Conocer los diferentes tipos de pruebas y la importancia del tester en el desarrollo de software <p>Genéricas:</p> <ul style="list-style-type: none"> ● Capacidad de administrar, organizar, planificar y liderar. ● Comunicación oral y escrita ● Habilidades en el estudio y manejo de las TI emergentes ● Toma de decisiones. ● Capacidad crítica y autocrítica ● Capacidad de trabajo en equipo ● Habilidades interpersonales ● Compromiso ético. ● Capacidad de aplicar los conocimientos en la práctica ● Habilidades de investigación ● Habilidad para trabajar en forma autónoma ● Búsqueda del logro. 	<ul style="list-style-type: none"> ● Seleccionar un artículo o documento introductorio sobre la importancia del tester en el desarrollo de software y guiar a los estudiantes a través de la lectura, haciendo preguntas y discutiendo los conceptos clave. ● Realizar una investigación sobre los diferentes tipos de pruebas y sus usos en el proceso de desarrollo de software.
2. Técnicas estáticas	
Competencias	Actividades de aprendizaje
<p>Específica:</p> <ul style="list-style-type: none"> ● Conocer técnicas y herramientas de pruebas estáticas para detectar y corregir errores en el código de un programa antes de su ejecución. 	<ul style="list-style-type: none"> ● Revisión de código en equipo: dividir a los estudiantes en equipos y asignarles un fragmento de código para revisar en busca de errores. ● Ejercicios prácticos: proporcionar a los estudiantes ejercicios prácticos en los que

<p>Genéricas:</p> <ul style="list-style-type: none"> ● Capacidad de administrar, organizar, planificar y liderar. ● Comunicación oral y escrita ● Habilidades en el estudio y manejo de las TI emergentes ● Toma de decisiones. ● Capacidad crítica y autocrítica ● Capacidad de trabajo en equipo ● Habilidades interpersonales ● Compromiso ético. ● Capacidad de aplicar los conocimientos en la práctica ● Habilidades de investigación ● Habilidad para trabajar en forma autónoma ● Búsqueda del logro. 	<p>deban aplicar las técnicas de pruebas estáticas para encontrar errores en el código.</p>
<p>3. Técnicas de diseño de pruebas</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica:</p> <ul style="list-style-type: none"> ● Conocer las diferentes técnicas y metodologías utilizadas para diseñar y ejecutar pruebas efectivas para software y sistemas. <p>Genéricas:</p> <ul style="list-style-type: none"> ● Capacidad de administrar, organizar, planificar y liderar. ● Comunicación oral y escrita ● Habilidades en el estudio y manejo de las TI emergentes ● Toma de decisiones. ● Capacidad crítica y autocrítica ● Capacidad de trabajo en equipo ● Habilidades interpersonales ● Compromiso ético. ● Capacidad de aplicar los conocimientos en la práctica 	<ul style="list-style-type: none"> ● Lectura y discusión de los conceptos clave y las técnicas de diseño de pruebas. ● Análisis de casos de estudio y ejemplos de pruebas reales en diferentes proyectos de software. ● La realización de prácticas y ejercicios prácticos para poner en práctica los conceptos aprendidos. ● Diseñar y ejecutar pruebas para un sistema desarrollado previamente en otra asignatura..

<ul style="list-style-type: none"> • Habilidades de investigación • Habilidad para trabajar en forma autónoma • Búsqueda del logro. 	
<p>4. Plan de pruebas</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica:</p> <ul style="list-style-type: none"> • Diseñar y planificar un plan de pruebas eficaz para un proyecto de software. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de administrar, organizar, planificar y liderar. • Comunicación oral y escrita • Habilidades en el estudio y manejo de las TI emergentes • Toma de decisiones. • Capacidad crítica y autocrítica • Capacidad de trabajo en equipo • Habilidades interpersonales • Compromiso ético. • Capacidad de aplicar los conocimientos en la práctica • Habilidades de investigación • Habilidad para trabajar en forma autónoma • Búsqueda del logro. 	<ul style="list-style-type: none"> • Investigar y analizar los diferentes métodos y herramientas utilizadas en la planificación de pruebas de software. • Identificar y analizar los requisitos y objetivos de un proyecto de software para determinar qué tipos de pruebas son necesarias. • Diseñar un plan de pruebas utilizando un enfoque estratégico y táctico. • Utilizar herramientas y técnicas para el seguimiento y el monitoreo del plan de pruebas. • Aplicar una metodología ágil para la planificación y ejecución de las pruebas.
<p>5. Herramientas de testing</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica:</p> <ul style="list-style-type: none"> • Conocer las diferentes herramientas utilizadas para el desarrollo de pruebas. <p>Genéricas:</p>	<ul style="list-style-type: none"> • Utilizar herramientas y técnicas para el seguimiento y el monitoreo del plan de pruebas. • Desarrollar un proyecto donde se aplique el diseño y planificación de un plan de

<ul style="list-style-type: none"> ● Capacidad de administrar, organizar, planificar y liderar. ● Comunicación oral y escrita ● Habilidades en el estudio y manejo de las TI emergentes ● Toma de decisiones. ● Capacidad crítica y autocrítica ● Capacidad de trabajo en equipo ● Habilidades interpersonales ● Compromiso ético. ● Capacidad de aplicar los conocimientos en la práctica ● Habilidades de investigación ● Habilidad para trabajar en forma autónoma ● Búsqueda del logro. 	<p>pruebas.</p> <ul style="list-style-type: none"> ● Instalar herramientas básicas utilizadas en los diferentes tipos de testing.
---	--

8. Práctica(s) (para fortalecer las competencias de los temas y de la asignatura)

<ul style="list-style-type: none"> ● Asignar un proyecto en equipo en el que los estudiantes deban aplicar las técnicas de pruebas estáticas para encontrar y corregir errores en un programa realizado previamente. ● Diseñar y ejecutar pruebas para un sistema desarrollado previamente en otra asignatura. ● Diseñar un plan de pruebas utilizando un enfoque estratégico y táctico. ● Desarrollar un proyecto donde se aplique el diseño y planificación de un plan de pruebas. ● Instalar herramientas básicas utilizadas en los diferentes tipos de testing.
--

9. Proyecto integrador (Para fortalecer las competencias de la asignatura con otras asignaturas)

<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"> ● 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 específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesional, 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 (específicas y genéricas de la asignatura)

Para evaluar las actividades de aprendizaje se recomienda solicitar: resúmenes, cuadros sinópticos, cuadro comparativo, informes, desarrollo de proyecto, reportes, estudio de casos, exposiciones en clase, reportes de visitas y portafolio de evidencias.

Para verificar el nivel del logro de las competencias del estudiante se recomienda utilizar: listas de cotejo, matrices de valoración, rúbricas, guías de observación, coevaluación y autoevaluación.

11. Fuentes de información (actualizadas considerando los lineamientos de la APA*)

1. Myers, G. J., Sandler, C., & Badgett, T. (2011). The art of software testing (3rd ed.). Hoboken, NJ: John Wiley & Sons.
2. Kaner, C., Bach, J., & Pettichord, B. (2002). Lessons learned in software testing: A context-driven approach. New York, NY: Wiley.
3. Beizer, B. (1990). Software testing techniques (2nd ed.). New York, NY: Van Nostrand Reinhold.
4. Black, R. (2003). Managing the testing process: Practical tools and techniques for managing hardware and software testing (2nd ed.). Indianapolis, IN: Wiley.
5. Fewster, M., & Graham, D. (1999). Software test automation: Effective use of test execution tools. Boston, MA: Addison-Wesley.