

1. DATOS GENERALES DE LA ASIGNATURA

Nombre de la asignatura:	Taller del testing
Clave de la asignatura:	TIT-2305
SATCA¹:	3-2-5
Carrera:	Ingeniería Informática

2. PRESENTACIÓN

Caracterización de la asignatura
<p>La asignatura cubrirá las diferentes herramientas y técnicas de testing como Selenium y JUnit, metodologías ágiles, etc.</p> <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 y usar las herramientas y técnicas adecuadas para probar una aplicación.

5. Competencias previas

<p>Las competencias previas necesarias para la asignatura Fundamentos de Testing incluyen:</p> <ul style="list-style-type: none"> ● Conceptos básicos en metodología de pruebas: Los estudiantes deben conocer los diferentes tipos de pruebas y la importancia del tester en el desarrollo de software ● 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	Pruebas unitarias	<ol style="list-style-type: none"> 1. Conceptos básicos 2. Objetivos y beneficios 3. Herramientas de las pruebas unitarias
2	Pruebas de integración	<ol style="list-style-type: none"> 1. Conceptos básicos 2. Objetivos y beneficios 3. Herramientas de pruebas de integración
3	Pruebas de aceptación	<ol style="list-style-type: none"> 1. Conceptos básicos 2. Objetivos y beneficios 3. Herramientas de pruebas de aceptación
4	Pruebas de rendimiento	<ol style="list-style-type: none"> 1. Conceptos básicos 2. Objetivos y beneficios 3. Herramientas de pruebas de rendimiento
5	Automatización de pruebas	<ol style="list-style-type: none"> 1. Diferencia entre automatización de pruebas y testing manual 2. Herramientas y técnicas populares de automatización de pruebas 3. Scripts de automatización de pruebas

7. Actividades de aprendizaje de los temas

1. Pruebas unitarias	
Competencias	Actividades de aprendizaje
<p>Específica:</p> <ul style="list-style-type: none"> ● Adquirir conocimientos técnicos necesarios para desarrollar pruebas unitarias para asegurar la calidad del código. <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 pruebas unitarias y guiar a los estudiantes a través de la lectura, haciendo preguntas y discutiendo los conceptos clave. ● Invitar a un especialista en pruebas unitarias para que demuestre cómo configurar y utilizar una herramienta para escribir y ejecutar pruebas unitarias. ● Escribir pruebas unitarias en un sistema desarrollado previamente para asegurar que el código funcione correctamente. ● Presentar los proyectos y cómo han implementado pruebas unitarias.
2. Pruebas de integración	
Competencias	Actividades de aprendizaje
<p>Específica:</p> <ul style="list-style-type: none"> ● Adquirir conocimientos técnicos necesarios para desarrollar pruebas de integración para asegurar que las diferentes partes de una aplicación funcionen correctamente juntas. <p>Genéricas:</p>	<ul style="list-style-type: none"> ● Seleccionar un artículo o documento introductorio sobre pruebas de integración y guiar a los estudiantes a través de la lectura, haciendo preguntas y discutiendo los conceptos clave. ● Invitar a un especialista en pruebas de integración para que demuestre cómo configurar y utilizar una herramienta para

<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>escribir y ejecutar pruebas de integración.</p> <ul style="list-style-type: none"> ● Escribir pruebas de integración en un sistema desarrollado previamente para asegurar que el código funcione correctamente. ● Presentar sus proyectos y cómo han implementado pruebas de integración.
<p>3. Pruebas de aceptación</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica:</p> <ul style="list-style-type: none"> ● Adquirir conocimientos técnicos necesarios para desarrollar pruebas de aceptación para asegurar que una aplicación cumpla con los requisitos del usuario. <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 	<ul style="list-style-type: none"> ● Seleccionar un artículo o documento introductorio sobre pruebas de aceptación y guiar a los estudiantes a través de la lectura, haciendo preguntas y discutiendo los conceptos clave. ● Invitar a un especialista en pruebas de aceptación para que demuestre cómo configurar y utilizar una herramienta como Cucumber para escribir y ejecutar pruebas de aceptación. ● Escribir pruebas de aceptación en un sistema desarrollado previamente para asegurar que una aplicación cumpla con estos requisitos. ● Presentar sus proyectos y cómo han implementado pruebas de aceptación.

<ul style="list-style-type: none"> ● Búsqueda del logro. 	
<p>4. Pruebas de rendimiento</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica:</p> <ul style="list-style-type: none"> ● Adquirir conocimientos técnicos necesarios para desarrollar pruebas de rendimiento para medir y optimizar el desempeño de una aplicación. <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 pruebas de rendimiento y guiar a los estudiantes a través de la lectura, haciendo preguntas y discutiendo los conceptos clave. ● Invitar a un especialista en pruebas de rendimiento para que demuestre cómo configurar y utilizar una herramienta para escribir y ejecutar pruebas de integración. ● Escribir pruebas de rendimiento en un sistema desarrollado previamente para asegurar que una aplicación cumpla con estos requisitos. ● Presentar sus proyectos y cómo han implementado pruebas de rendimiento.
<p>5. Automatización de pruebas</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica:</p> <ul style="list-style-type: none"> ● Adquirir conocimientos técnicos necesarios para desarrollar scripts de automatización de pruebas de forma eficiente utilizando herramientas populares. 	<ul style="list-style-type: none"> ● Investigar sobre diferentes herramientas y técnicas populares de automatización de pruebas (Selenium, JUnit, etc.) y su uso en diferentes tipos de pruebas (unitarias, de integración, de aceptación, de rendimiento, etc.). ● Practicar la creación de scripts de

<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>automatización de pruebas utilizando diferentes lenguajes de programación (Python, Java, etc.) y herramientas populares (Selenium, JUnit, etc.).</p>
---	---

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

<ul style="list-style-type: none"> ● Escribir pruebas unitarias en un sistema desarrollado previamente para asegurar que el código funcione correctamente. ● Escribir pruebas de integración en un sistema desarrollado previamente para asegurar que el código funcione correctamente. ● Escribir pruebas de aceptación en un sistema desarrollado previamente para asegurar que una aplicación cumpla con estos requisitos. ● Escribir pruebas de rendimiento en un sistema desarrollado previamente para asegurar que una aplicación cumpla con estos requisitos.
--

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. Beck, K. (2002). Test-driven development: By example. Addison-Wesley Professional.
2. Fowler, M. (2005). Refactoring: Improving the design of existing code. Addison-Wesley Professional.
3. Spolsky, J. (2004). Joel on software: And on diverse and occasionally related matters that will prove of interest to software developers, designers, and managers, and to those who, whether by good fortune or ill luck, work with them in some capacity. Apress.
4. Graham, R. (2008). Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality. Addison-Wesley Professional.
5. Fewster, M., & Graham, D. (2013). Software test automation: Effective use of test execution tools. Addison-Wesley Professional.