	Nombre del documento: Formato de Programa de Estudio de asignatura de Especialidad	Código: TecNM-AC-PO-007-02
		Revisión: 0
	Referencia a la Norma ISO 9001:2015 8.3, 8.3.1	Página 1 de 10


1. Datos Generales de la asignatura

Nombre de la asignatura:	Integración de proyectos de TI
Clave de la asignatura:	TAB-2406
SATCA¹:	1 - 4 - 5
Carrera:	Ingeniería Informática

2. Presentación

Caracterización de la asignatura
<p>Contribuye al perfil del Ingeniero en Informática al proporcionar conocimientos especializados, habilidades técnicas específicas en el uso de las tecnologías actuales y emergentes para la integración e implementación de soluciones aplicables a múltiples entornos de hardware y software para dar solución a diversas problemáticas de su contexto social y tecnológico, permitiendo su desempeño profesional y ético, cumpliendo con los estándares de vanguardia en las organizaciones.</p> <p>La asignatura se distingue por su enfoque práctico y la aplicación integral de conocimientos para integrar soluciones tecnológicas a problemas del contexto social y tecnológico.</p>
Intención didáctica
<p>Se organiza el temario, agrupando los contenidos de la asignatura en cuatro temas de manera secuencial, distribuyendo los conceptos teóricos con actividades prácticas para el adecuado entendimiento e interpretación de las prácticas que se realizarán a lo largo del curso lo cual permitirá el óptimo desarrollo y alcance de competencias que esta asignatura proporciona.</p> <p>En la unidad 1 se instruye a los estudiantes en los principios fundamentales de desarrollo y gestión de proyectos de software, proporcionando una comprensión integral de los conceptos, herramientas y técnicas necesarias para llevar a cabo proyectos exitosos en este campo.</p> <p>Para la unidad 2 proporcionar a los estudiantes una comprensión integral de las pruebas unitarias en el desarrollo de software, así como las mejores prácticas y herramientas para su implementación efectiva y su integración en procesos de integración continua identificando y analizando las ventajas que ofrecen las pruebas unitarias en términos de detección temprana de errores, mejora de la calidad del código y facilitación del proceso de depuración así como explorar las mejores prácticas y principios fundamentales para</p>

¹ Sistema de Asignación y Transferencia de Créditos Académicos

	Nombre del documento: Formato de		Código: TecNM-AC-PO-007-02
	Programa de Estudio de asignatura de Especialidad		Revisión: 0
	Referencia a la Norma ISO 9001:2015	8.3, 8.3.1	Página 2 de 10

escribir pruebas unitarias efectivas, como la independencia, la reproducibilidad, la rapidez y la claridad.

Entender el concepto de integración continua y su importancia en el desarrollo ágil de software.

En la unidad 3 se pretende proporcionar a los estudiantes una comprensión profunda de las pruebas de integración en el desarrollo de software y familiarizarlos con las herramientas y tecnologías utilizadas en este proceso. Explorando diferentes estrategias de prueba de integración, incluyendo pruebas de componentes individuales, servicios y pruebas de extremo a extremo así como comprender cómo seleccionar y aplicar la estrategia de prueba adecuada según los requisitos y características del proyecto.

En la unidad 4 se busca Capacitar a los estudiantes en el uso efectivo de herramientas y metodologías en el desarrollo de software, centrándose en el desarrollo guiado por pruebas (TDD), herramientas de pruebas para diferentes plataformas, herramientas de integración continua (CI) y herramientas para implementación de proyectos de tecnologías de la información (TI).

Comprender los principios fundamentales del desarrollo guiado por pruebas (TDD) y su impacto en la calidad del software y practicar la aplicación de TDD en la escritura de código mediante ejercicios prácticos y proyectos de aplicación. Familiarizarse con Jenkins como una herramienta de automatización de CI de código abierto ampliamente utilizada en el desarrollo de software y Comprender cómo utilizar estas herramientas para automatizar el despliegue y la configuración de infraestructuras y aplicaciones en entornos de producción.

3. Participantes en el diseño y seguimiento curricular del programa


Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico Superior de Huauchinango, mayo 2024.	Academia Ingeniería Informática.	Reuniones para la elaboración de la Especialidad.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Genera estrategias de gestión e integración de Tecnologías apoyado en marcos de calidad orientados a generar productos y/o servicios de valor agregado a proyectos soportados en el uso de TI.

5. Competencias previas


Conocer conceptos de desarrollo de sw, administración de proyectos, también conocer y haber desarrollado distintas pruebas al sw conociendo los distintos tipos de pruebas a
--

	Nombre del documento: Formato de	Código: TecNM-AC-PO-007-02
	Programa de Estudio de asignatura de Especialidad	Revisión: 0
	Referencia a la Norma ISO 9001:2015 8.3, 8.3.1	Página 3 de 10

aplicar. También debe conocer básicamente las tecnologías emergentes utilizadas para proyectos de TI.

6. Temario


No.	Temas	Subtemas
1	Gestión y Calidad de proyectos de TI	1.1 Conceptos generales de Proyecto 1.1.1 Concepto e importancia 1.1.2 Planificación y seguimiento 1.2 Importancia de la gestión e integración de proyectos de TI. 1.3 Calidad y Pruebas de sw 1.3.1 Modelo de pruebas de sw V y W 1.4 Integración continua (CI) y DevOps 1.4.1 Beneficios de la CI 1.4.2 Herramientas de CI
2	Pruebas Unitarias	2.1 Pruebas Unitarias (conceptos y Ventajas) 2.2 Mejores prácticas para escribir pruebas unitarias efectivas 2.3 Herramientas para automatizar pruebas unitarias 3.3.1 Instalación y configuración 2.4 Integración continua (CI) 3.4.1 Integración de pruebas unitarias a la integración continua 2.5 Uso de mocks, stubs y fakes en pruebas unitarias
3	Pruebas de Integración	3.1 Pruebas Integración (Conceptos y ventajas) 3.2 Ejecución Manual vs Automatizada 3.3 Diseño Caso de Prueba de Integración 3.4 Estrategias de prueba de Integración (componentes, servicios, de extremo a extremo) 3.5 Herramientas y tecnologías para pruebas de integración 3.5.1 Frameworks de automatización de pruebas (Selenium, Cypress, etc) 3.5.2 Herramientas de virtualización de servicios (WireMock, MockServer, etc) 3.5.3 Herramientas de gestión de contenedores (Docker, Kubernetes, etc).

	Nombre del documento: Formato de		Código: TecNM-AC-PO-007-02
	Programa de Estudio de asignatura de Especialidad		Revisión: 0
	Referencia a la Norma ISO 9001:2015	8.3, 8.3.1	Página 4 de 10


4	Automatización de Procesos	4.1 TDD Desarrollo Guiado por Pruebas. 4.2 Herramientas de pruebas para diferentes plataformas (por ejemplo, aplicaciones móviles, aplicaciones web). 4.3 Herramienta de automatización de código abierto para la integración continua (CI) Jenkins. 4.4 Herramienta para implementación de proyectos TI.
---	----------------------------	--

7. Actividades de aprendizaje de los temas


1. Gestión y Calidad de Proyectos de TI	
Competencias	Actividades de aprendizaje
<p>Específica(s): Comprender los conceptos generales de gestión de proyectos y su importancia en el contexto de las tecnologías de la información.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> Habilidad para buscar y analizar información proveniente de fuentes diversas. Capacidad de análisis y síntesis. Trabajo en equipo. Capacidad de aplicar los conocimientos. Habilidades de investigación. Capacidad de generar nuevas ideas. Búsqueda del logro. 	<ul style="list-style-type: none"> En equipos crear un plan de proyecto ficticio para desarrollar una aplicación móvil o web. Incluir elementos como objetivos, alcance, cronograma, recursos. Realizar en clase un estudio de caso de fracaso de proyectos de TI identificando casos reales de proyectos de TI que fracasaron debido a una mala gestión o falta de integración. Discutir en clase sobre las lecciones aprendidas y cómo podrían haberse evitado esos fracasos. Diseñar casos de prueba para cubrir diferentes escenarios de uso. En equipo de trabajo los estudiantes deben implementar un modelo de pruebas V o W en un proyecto de software específico. Investigar y elaborar una lista de los principales beneficios de la integración continua. Generar una guía de la configuración de una herramienta de integración continua como Jenkins o GitLab CI.
2. Pruebas Unitarias	
Competencias	Actividades de aprendizaje

	Nombre del documento: Formato de	Código: TecNM-AC-PO-007-02
	Programa de Estudio de asignatura de Especialidad	Revisión: 0
	Referencia a la Norma ISO 9001:2015 8.3, 8.3.1	Página 5 de 10

<p>Específica(s):</p> <p>Entender los principios fundamentales de la calidad del software y las estrategias de prueba.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Capacidad de aplicar los conocimientos. • Liderazgo. • Búsqueda del logro. 	<ul style="list-style-type: none"> • Escribir pruebas unitarias para una función o módulo específico. Luego, en sesión revisar e intercambiar las pruebas unitarias proporcionando retroalimentación constructiva. • Revisar y analizar pruebas unitarias mal escritas. Revisar en clase solicitando que trabajen en parejas para identificar y corregir los problemas, aplicando las mejores prácticas aprendidas en clase. • Instalar y configurar alguna herramienta de pruebas unitarias como JUnit, NUnit o PHPUnit en sus entornos de desarrollo. • Desarrollar en pruebas unitarias que escriban y ejecuten utilizando la herramienta recién instalada. • Configure un pipeline de integración continua utilizando una herramienta como Jenkins o GitLab CI. • Los estudiantes deben realizar cambios en el código y observar cómo se ejecutan las pruebas unitarias como parte del proceso de integración continua. • Escribir pruebas unitarias para la función o clase utilizando mocks para simular el comportamiento de las dependencias externas. Discutir en clase cómo el uso de mocks puede mejorar la modularidad y la velocidad de las pruebas unitarias. •
3. Pruebas de Integración	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Conocer los conceptos y herramientas relacionados con la integración continua y las pruebas de unidad.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. 	<ul style="list-style-type: none"> • Investigar sobre las ventajas y desventajas de las pruebas de integración. Presentar en clase un debate donde se presenten sus hallazgos y argumente a favor o en contra de las pruebas de integración. • Analizar casos de estudio de

	Nombre del documento: Formato de	Código: TecNM-AC-PO-007-02
	Programa de Estudio de asignatura de Especialidad	Revisión: 0
	Referencia a la Norma ISO 9001:2015 8.3, 8.3.1	Página 6 de 10


<ul style="list-style-type: none"> • Capacidad de organizar y planificar. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Trabajo en equipo. • Capacidad de aplicar los conocimientos. • Habilidades de investigación. • Capacidad de generar nuevas ideas. • Liderazgo. • Habilidad para trabajar en forma autónoma. • Búsqueda del logro. 	<p>proyectos que han obtenido beneficios significativos al implementar pruebas de integración y presentar sus conclusiones sobre cómo las pruebas de integración contribuyeron al éxito del proyecto.</p> <ul style="list-style-type: none"> • Analizar escenarios: De ejecución manual de pruebas de integración y para automatizar para comparar los pros y los contras de cada método, discutiendo sobre la eficiencia, la confiabilidad y la escalabilidad. • Analizar casos de integración simples e implementarlos en herramientas de automatización como Selenium y Cypress. • Diseñar casos de prueba de integración (en un servicio web o móvil) que cubran diferentes aspectos del escenario, como la funcionalidad, la seguridad y el rendimiento. En clase revise los casos de prueba donde los estudiantes intercambien y revisen los casos de prueba desarrollados por sus compañeros. • Desarrollar una simulación de pruebas de extremo a extremo. • Investigar y presentar hallazgos y recomendaciones de uso sobre herramientas y tecnologías para pruebas de integración Selenium, WireMock y Docker. <p>Implementar y ejecutar pruebas de integración en el entorno de contenedores, utilizando herramientas como WireMock para virtualizar servicios.</p>
---	---

	Nombre del documento: Formato de	Código: TecNM-AC-PO-007-02
	Programa de Estudio de asignatura de Especialidad	Revisión: 0
	Referencia a la Norma ISO 9001:2015 8.3, 8.3.1	Página 7 de 10


4. Automatización de Procesos	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Aplicar estrategias de prueba de integración y explorar herramientas para el desarrollo guiado por pruebas.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Trabajo en equipo. • Capacidad de aplicar los conocimientos. • Habilidades de investigación. • Capacidad de generar nuevas ideas. • Liderazgo. • Habilidad para trabajar en forma autónoma. • Búsqueda del logro. 	<ul style="list-style-type: none"> • Instalar y configurar en clase Jenkins en un entorno de desarrollo local o en un servidor. • Desafíe a los estudiantes a diseñar y desarrollar un pipeline completo de CI/CD utilizando Jenkins para un proyecto de software real o ficticio. • Investigar y evaluar diferentes herramientas de implementación de proyectos de TI, como Ansible, Puppet o Chef. Presentar sus hallazgos y recomendaciones, destacando características clave, casos de uso y consideraciones de implementación. • Desarrollar un escenario de implementación para un proyecto de TI, como la configuración de un servidor web o la implementación de una base de datos.

8. Práctica(s)

<ul style="list-style-type: none"> • Elaboración de un plan de proyecto • Automatización de Despliegues. Utilizar herramientas como Jenkins o GitLab CI para automatizar el proceso de construcción y despliegue del software. • Automatización de Despliegues. Utilizar herramientas como Jenkins o GitLab CI para automatizar el proceso de construcción y despliegue del software. • Control de Versiones. Implementa un sistema de control de versiones como Git para gestionar el código fuente y facilitar la colaboración entre miembros del equipo. • Implementación de Pruebas Unitarias con Herramientas de Automatización: Seleccionar herramientas para pruebas unitarias que se ajuste a tu proyecto, como JUnit para Java, NUnit para .NET, o Jest para JavaScript.
--

	Nombre del documento: Formato de		Código: TecNM-AC-PO-007-02
	Programa de Estudio de asignatura de Especialidad		Revisión: 0
	Referencia a la Norma ISO 9001:2015	8.3, 8.3.1	Página 8 de 10

- Ejecución Automatizada: Configurar el sistema de integración continua (CI) para que ejecute automáticamente las pruebas unitarias cada vez que se realice un cambio en el código base.
- Implementación de Pruebas de Integración Automatizadas:
Selección de Framework: Investigar y seleccionar un framework de automatización de pruebas adecuado para pruebas de integración, como Selenium para pruebas de interfaz de usuario web o RestAssured para pruebas de servicios REST. Identificar y diseñar escenarios de prueba de integración que abarquen la interacción entre componentes, servicios o sistemas de extremo a extremo.
- Implementación de Pruebas Automatizadas. Utilizar el framework seleccionado para implementar pruebas de integración automatizadas que verifiquen el funcionamiento correcto de la integración entre los diferentes componentes o servicios. Configurar la ejecución automatizada de las pruebas de integración como parte del flujo de trabajo de integración continua (CI) y monitorear los resultados para identificar posibles problemas de integración.
- Desarrollo Guiado por Pruebas (TDD) para una Aplicación Web utilizando Jenkins como parte de la Integración Continua (CI)
- Selección de Tecnología y Herramientas. Definir el tipo de aplicación web que desarrollarás y las tecnologías que utilizarás (por ejemplo, React para el frontend, Node.js para el backend). Selecciona las herramientas de prueba adecuadas para tu aplicación web, como Jest para pruebas unitarias, Selenium para pruebas de interfaz de usuario, y Appium para pruebas móviles. Configuración de Entorno de Desarrollo y asegúrate de que todas las herramientas de prueba estén instaladas y configuradas correctamente. Implementar la funcionalidad mínima necesaria en tu aplicación para que la prueba pase refactorizar el código
- Configurar un servidor Jenkins en tu entorno de desarrollo o en un servidor remoto y configurar un nuevo proyecto en Jenkins para tu aplicación web, configurando el repositorio de código fuente y definiendo los pasos de compilación y prueba. Integrar pruebas automatizadas en el proceso de compilación de Jenkins para que se ejecuten automáticamente cada vez que se realice un cambio en el repositorio.
- Implementación Continua. Configurar Jenkins para que despliegue automáticamente su aplicación web en un entorno de prueba o producción después de que las pruebas pasen exitosamente.
- Utiliza herramientas como Docker y Kubernetes u otras para facilitar la implementación y administración de tu aplicación en entornos de producción.

	Nombre del documento: Formato de		Código: TecNM-AC-PO-007-02
	Programa de Estudio de asignatura de Especialidad		Revisión: 0
	Referencia a la Norma ISO 9001:2015	8.3, 8.3.1	Página 9 de 10

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 comunitaria, 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-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

Para evaluar las actividades de aprendizaje, se recomienda solicitar: Reportes de prácticas, estudios de casos, exposiciones en clase, ensayos, portafolio de evidencias, cuadros comparativos.

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

	Nombre del documento: Formato de		Código: TecNM-AC-PO-007-02
	Programa de Estudio de asignatura de Especialidad		Revisión: 0
	Referencia a la Norma ISO 9001:2015	8.3, 8.3.1	Página 10 de 10

11. Fuentes de información

- Valdés, M. (2018). Gestión de Proyectos. Ediciones Universidad Alberto Hurtado.
- Pérez, R. (2018). Mocks, Stubs y Fakes: Técnicas para Pruebas de Software. Editorial UOC.
- Ramírez, C. (2019). Pruebas Unitarias en Desarrollo de Software. Ediciones Anaya Multimedia.
- Gómez, P. (2018). Desarrollo Ágil y Pruebas Unitarias. Ediciones Paraninfo.
- Pérez, E. (2018). Diseño de Casos de Prueba: Métodos y Herramientas. Marcombo.
- Martínez, J. (2019). Pruebas de Integración con Selenium y Docker. Ediciones Universidad Politécnica de Madrid.
- Fernández, M. (2019). Desarrollo Guiado por Pruebas: TDD en Acción. Editorial UOC.
- Martínez, R. (2017). Jenkins: Automatización de la Integración Continua. Editorial Ra-Ma.
- Pérez, E. (2018). Implementación de Proyectos de Tecnologías de la Información. Ediciones Paraninfo.