

1. DATOS GENERALES DE LA ASIGNATURA

Nombre de la asignatura:	DevOps
Clave de la asignatura:	TIF-2302
SATCA¹:	3-2-5
Carrera:	Ingeniería Informática

2. PRESENTACIÓN

Caracterización de la asignatura
La asignatura de DevOps se enfoca en el desarrollo e implementación de prácticas y herramientas para mejorar la colaboración entre los equipos de desarrollo y operaciones, con el objetivo de lograr un ciclo de vida de software más rápido y eficiente. La caracterización de esta asignatura incluye: • Introducción al concepto de DevOps y sus principios fundamentales, como la automatización, la colaboración y la entrega continua. • Conocimiento de herramientas y tecnologías utilizadas en DevOps, como Git, Jenkins, Ansible, Docker y Kubernetes. • Habilidad para implementar un sistema de integración y entrega continua (CI/CD) utilizando herramientas como Jenkins o Travis CI. • Habilidad para desplegar y escalar aplicaciones en entornos de contenedores, utilizando herramientas como Docker o Kubernetes. • Conocimiento de las tendencias y desarrollos actuales en DevOps y su capacidad para evaluar su relevancia para diferentes escenarios de negocio. Habilidad para trabajar en equipos multidisciplinarios y colaborar con diferentes áreas de una organización para lograr un ciclo de vida de software más eficiente.
Intención didáctica
La intención didáctica de la materia DevOps es desarrollar en los estudiantes las habilidades y conocimientos necesarios para implementar y administrar prácticas y herramientas de DevOps en una organización. Esto incluye: • Clases teóricas: se proporcionarán conocimientos fundamentales sobre los conceptos de DevOps.

¹ Sistema de Asignación y Transferencia de Créditos Académicos

- Prácticas en el laboratorio: los estudiantes tendrán la oportunidad de poner en práctica los conceptos aprendidos en el curso mediante la implementación de soluciones en la nube utilizando herramientas y plataformas en la nube.
- Proyectos: los estudiantes completarán proyectos individuales o en grupo para poner en práctica los conceptos aprendidos en el curso.
- Tareas y evaluaciones: se asignarán tareas y evaluaciones para medir el progreso y el logro de los objetivos del curso.
- Discusiones en clase y trabajo en grupo: se fomentará la discusión y el intercambio de ideas entre los estudiantes a través de discusiones en clase y trabajo en grupo.

En general, la intención didáctica de la materia de Computación en la Nube es proporcionar a los estudiantes una comprensión completa de los conceptos y tecnologías relacionadas con la computación en la nube, y capacitarlos para diseñar, implementar y administrar soluciones escalables y seguras en la nube utilizando una variedad de plataformas y herramientas.

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
Una competencia a desarrollar en un curso de DevOps es la habilidad para implementar y administrar un sistema de integración y entrega continua (CI/CD) utilizando herramientas y tecnologías DevOps. Esto incluiría la capacidad de configurar y utilizar herramientas como Jenkins o Travis CI para automatizar el proceso de compilación, pruebas y despliegue de aplicaciones, así como la capacidad de monitorear y optimizar el rendimiento del sistema.

5. Competencias previas

Para un curso de DevOps, se requieren las siguientes competencias previas:

- Conocimientos básicos de sistemas operativos y redes, incluyendo la comprensión de cómo funcionan y cómo se administran.
- Conocimientos básicos de programación, ya que se requiere para automatizar procesos y construir scripts.
- Conocimientos previos en sistemas de control de versiones, como Git, ya que es una herramienta esencial en DevOps.
- Conocimientos previos en infraestructura y administración de servidores, ya que se requiere para implementar y administrar sistemas de integración y entrega continua (CI/CD).
- Conocimientos previos en seguridad informática, ya que se requiere para garantizar la seguridad de las aplicaciones y servicios desplegados en producción.
- Conocimientos previos en metodologías ágiles, ya que DevOps es una filosofía de trabajo ágil.
- Conocimientos previos en la implementación de contenedores, ya que son una herramienta esencial en DevOps.

6. Temario

No.	Temas	Subtemas
1	Introducción a DevOps	1. Conceptos básicos 2. Historia, principios 3. Objetivos de DevOps.
2	Gestión de la configuración	1. Introducción 2. Control de Configuración 3. Herramientas de gestión de configuración
3	Integración continua	1. Buenas prácticas 2. Beneficios 3. Software de Integración Continua
4	Testing agile	1. Conceptos básicos (cuadrantes, pirámide del testing agile) 2. Prácticas de testing agile 3. Automatización de pruebas
5	Entrega continua	1. Introducción 2. Estrategias de despliegue 3. Herramientas de despliegue de software

7. Actividades de aprendizaje de los temas

1. Introducción a DevOps	
Competencias	Actividades de aprendizaje
Específica: - Comprender los conceptos básicos y objetivo de DevOps Genéricas: - 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.	Seleccionar un artículo o documento introductorio sobre DevOps y guiar a los estudiantes a través de la lectura, haciendo preguntas y discutiendo los conceptos clave.
2. Gestión de la configuración	
Competencias	Actividades de aprendizaje
Específica: Comprender los beneficios y objetivos de la gestión de configuración, así como los principios y prácticas necesarios para implementarla de manera efectiva. Genéricas:	- Estudio de casos: estudiar diferentes casos de uso de la gestión de configuración en diferentes industrias, como la tecnología, la banca, la salud, etc. - Prácticas de laboratorio: realizar prácticas de laboratorio utilizando diferentes herramientas de gestión de configuración,

• 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.	como Ansible, Puppet y Chef, para automatizar la configuración de servidores y aplicaciones.
3. Integración continua	
Competencias	Actividades de aprendizaje
Específica: • Comprender y aplicar los principios de integración continua en un entorno de desarrollo de software. Genéricas: • 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	• Organizar a los estudiantes en grupos pequeños y pedirles que discutan las ventajas y desventajas de la integración continua y cómo se relaciona con el proceso de desarrollo de software en general. • Investigar diferentes herramientas específicas relacionadas con la integración continua (como Jenkins, Travis CI o CircleCI, etc) y presenten sus hallazgos a la clase.

- Habilidad para trabajar en forma autónoma - Búsqueda del logro.	
4. Testing agile	
Competencias	Actividades de aprendizaje
Específica: - Comprender los principios de testing en un entorno de desarrollo ágil. Genéricas: - 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.	- Proporcionar a los estudiantes una introducción a los conceptos básicos de testing en un entorno ágil. - Pedir a los estudiantes que investiguen una herramienta de testing específica (como Selenium, JUnit o Cucumber) y presenten sus hallazgos a la clase.
5. Entrega continua	
Competencias	Actividades de aprendizaje
Específica:	Proporcionar a los estudiantes un ejemplo de una aplicación y pedirles que diseñen un pipeline de entrega continua para

● Implementar y utilizar un pipeline de entrega continua para automatizar el proceso de despliegue de una aplicación. Genéricas: ● 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.	automatizar el proceso de despliegue.
---	--

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

● Realizar prácticas de laboratorio utilizando diferentes herramientas de gestión de configuración, como Ansible, Puppet y Chef, para automatizar la configuración de servidores y aplicaciones. ● Realizar un proyecto que requiera que los estudiantes apliquen los conceptos de integración continua en un entorno real. ● Implementar un pipeline de entrega continua para automatizar el proceso de despliegue.
--

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

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 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*)

Guijarro Olivares, J. Caparrós Ramírez, J. & Cubero Luque, L. (2019). DevOps y seguridad cloud.. Editorial UOC. <https://elibro.net/es/lc/itsa/titulos/128889>