



1. Datos Generales de la asignatura

Nombre de la asignatura:	Cómputo en la nube II
Clave de la asignatura:	GED-2405
SATCA¹:	2-3-5
Carrera:	Ingeniería en Tecnologías de la Información y Comunicaciones e Ingeniería en Sistemas Computacionales.

2. Presentación

Caracterización de la asignatura

La asignatura de Cómputo en la nube II está diseñada para proporcionar al perfil del ingeniero los conocimientos y habilidades necesarios para comprender, implementar y gestionar eficazmente el ciclo de vida completo del desarrollo y operación de software mediante la metodología DevOps. DevOps, que fusiona el desarrollo (Dev) y las operaciones (Ops), se centra en mejorar la colaboración, la comunicación y la eficiencia en el desarrollo de software.

Cómputo en la nube II consiste en comprender los principios de DevOps, conocer las herramientas y tecnologías utilizadas en DevOps, así como desarrollar habilidades para la automatización de procesos e implementar prácticas de monitoreo.

Esta asignatura requiere que se tengan conocimientos de desarrollo de software, sistemas operativos y redes. Además, se beneficiarán de tener experiencia práctica en al menos una plataforma de desarrollo y conocimientos básicos de administración de sistemas.

Intención didáctica

El temario está organizado en cuatro temas:

El primer tema aborda las primeras dos etapas del ciclo DevOps que son la planeación y codificación, resaltando la importancia en la selección de la metodología a utilizar y la administración del código fuente de un proyecto.

El tema dos resalta la importancia de la elaboración de pruebas y la retroalimentación, en el desarrollo de proyectos que requiere habilidades para la automatización de procesos, la integración y entrega continua.

El tercer tema aborda los temas relacionados con la implementación y puesta en marcha del código elaborado, aquí la prioridad son los temas relacionados al manejo y administración de las versiones, tema primordial en los grandes proyectos.

¹ Sistema de Asignación y Transferencia de Créditos Académicos



Por último, el tema cuatro se encarga de relacionar el proyecto con la gestión de IaaS y la PaaS, la orquestación de contenedores y la retroalimentación de la UX, con la finalidad de cerrar el ciclo DevOps.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Tecnológico Nacional de México campus San Juan del Río. Diciembre del 2023.	I.S.C. Leonardo Valdes Arteaga I.S.C. Wilfrido Medina Varela I.S.C. Govani Gregorio Sánchez Orduña M.G.T.I. Rogelio Vázquez Nieves I.S.C. José Jaime Rodríguez Pedraza L.S.C. Domingo Rosales Álvarez	Reunión de Academia para el diseño de Módulo de Especialidad y elaboración de programas de estudio.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Implementar el ciclo DevOps para la elaboración de proyectos de cómputo en la nube, aplicando de manera estratégica cada una de las etapas en la integración y entrega continua de los servicios de TI de la organización

5. Competencias previas

Diseña e implementa soluciones de aplicaciones informáticas para servicios de computación en la nube como un nuevo modelo de prestación de servicios de negocio y tecnologías. Identifica, analiza y compara la diversidad de métodos, herramientas y criterios para la toma de decisiones en las organizaciones. Identifica y aplica los elementos que integran la aplicación CRM en el desarrollo de sistemas que reflejen la importancia de la relación con los clientes.

6. Temario

No.	Temas	Subtemas
1	Planeación y Codificación en DevOps	- Planeación - Selección de herramientas y metodologías orientadas a DevOps - Codificación. - Revisión de código fuente de un proyecto orientado a integración continua



		2.2. Administración de código fuente de un proyecto
2	Construcción, Compilación y Pruebas	1. Construcción o compilación 1.1. Herramientas de construcción, compilación y empaquetamiento estáticas. 1.2. Herramientas de construcción, compilación y empaquetamiento continuas. 2. Pruebas. 2.1. Importancia de las pruebas en el desarrollo de software 2.2. Tipos de pruebas 2.2.1. Pruebas unitarias 2.2.2. Pruebas Comportamiento 2.3. Retroalimentación
3	Liberación y Despliegue	3. Liberación 3.1. Repositorio de artefactos 3.2. Dependencias y bibliotecas 3.3. Registro, control de cambios y versiones 4. Despliegue 4.1. Configuración y creación de la imagen del contenedor 4.2. Publicación de la imagen
4	Operación y Monitoreo	5. Operación 5.1. Gestión de la IaaS o la PaaS 5.2. Orquestación de contenedores 6. Monitoreo 6.1. Herramientas de monitoreo 6.2. Retroalimentación de la experiencia de usuario.

7. Actividades de aprendizaje de los temas

1. Planeación y codificación en DevOps	
Competencias	Actividades de aprendizaje
Específica(s): Capacidad de implementar soluciones de monitoreo, recopilar métricas y realizar análisis para mejorar el rendimiento y la fiabilidad de las aplicaciones entre los equipos de desarrollo y operaciones, para lograr la entrega continua y confiable de software de alta calidad. Genéricas: 1. Conocimiento de desarrollo de software. 2. Familiaridad con metodologías ágiles. Familiaridad con metodologías ágiles. 3. Habilidades de gestión de proyectos. 4. Comprensión de sistemas operativos y redes. 5. Habilidades de colaboración y comunicación. 6. Mentalidad analítica.	• Generar un reporte para identificar las características del uso y entornos con DevOps. • Identificar, las herramientas para Integrar y probar el código de forma continua para detectar errores temprano y garantizar una calidad constante del código. • Identificar las herramientas para codificación, monitoreo y registro sobre el rendimiento, la disponibilidad y la confiabilidad del software en producción para detectar problemas y realizar mejoras proactivas.
2. Construcción, compilación y pruebas	
Competencias	Actividades de aprendizaje
Específica(s): Capacidad para automatizar eficientemente la construcción del código fuente, gestionar dependencias, ejecutar pruebas automatizadas y garantizar la calidad del software mediante el uso de herramientas de construcción, integración continua (CI), y despliegue continuo (CD). Genéricas: • Capacidad de análisis y síntesis. • Habilidades de gestión de información (habilidad para buscar y	• Hacer uso de herramientas de control de versiones como Git para gestionar el código fuente. • Realizar demostraciones en tiempo real de cómo funciona el proceso de compilación. • Utilizar herramientas de depuración para mejorar las habilidades de solución de problemas.

analizar información proveniente de fuentes diversas). - Habilidades de investigación. - Capacidad de aprender. - Capacidad de adaptarse a nuevas situaciones	- Realizar ejercicios prácticos donde se desarrollen y ejecuten prueba unitarias y comportamiento. - Introducir a los estudiantes en herramientas de automatización de pruebas.
3. Liberación y despliegue	
Competencias	Actividades de aprendizaje
Específica(s): Recopilar artefactos y empaquetar aplicativos para su posterior despliegue en una infraestructura Genéricas: - Habilidades de investigación. - Capacidad de aprender. - Capacidad de adaptarse a nuevas situaciones. - Conocimiento de una segunda lengua. - Comprensión de sistemas operativos y redes - Conocimiento de desarrollo de software	- Seleccionar un servidor de automatización y empaquetamiento. - Seleccionar un orquestador de contenedores o vm para el despliegue del aplicativo. - Ejecutar en una IaaS o PaaS con el orquestador seleccionado.
4. Operación y monitoreo	
Competencias	Actividades de aprendizaje
Específica(s): Identificar áreas de oportunidad, para mejorar las aplicaciones en base a la experiencia de usuario. Genéricas: - Capacidad de análisis y síntesis. - Conocimiento de una segunda lengua. - Habilidades de investigación. - Capacidad de aprender. - Habilidades de gestión de información (habilidad para buscar y analizar información proveniente de fuentes diversas).	- Investiga sobre al menos tres herramientas de monitoreo en la nube. Pueden ser herramientas populares como Prometheus, Grafana, Datadog, AWS CloudWatch, entre otras para comparar en términos de características, capacidades y ventajas - Investiga sobre al menos dos herramientas de análisis y retroalimentación de usuarios, sobre el comportamiento de los usuarios en las aplicaciones web como (hotjar, matomo, mouseflow, inspectlet entre otros),



- Búsqueda del logro.

8. Práctica(s)

La elaboración y diseño de las prácticas en la asignatura, es muy importante porque permite dar seguimiento a las instrucciones, seguir los planteamientos teóricos, implementar lo visto en el aula de clase y desarrollar las habilidades para complementar las competencias de los alumnos.

A continuación, se sugieren algunas:

- Implementar un proyecto de alguna materia previa como lo es Programación Web en el ciclo de DevOps.
 - Manejar un sistema de control de versiones.
 - Utilizar una herramienta multiplataforma de generación o automatización de código.
 - Utilizar una herramienta de desarrollo continuo / integración continua.
 - Seleccionar un orquestador (contenedores / máquinas virtuales).
- Comparar herramientas de construcción o automatización de código.



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

Son las técnicas, instrumentos y herramientas sugeridas para constatar los desempeños académicos de las actividades de aprendizaje.

Los procedimientos sugeridos son:

- Tareas y participación activa en clases
- Exámenes para evaluar la comprensión de los temas
- Reportes de prácticas
- Portafolio de evidencias
- Casos de estudios
- Reportes de prácticas

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.

11. Fuentes de información

Noah Gift. (2020) *Python for Devops: Learn Ruthlessly Effective Automation*. Editorial O'Reilly Media

Mikael Krief. (2022) *Learning DevOps - Second Edition: A comprehensive guide to accelerating DevOps culture adoption with Terraform, Azure DevOps, Kubernetes, and Jenkins*. Packt Publishing

Justin Domingus (2022) *Cloud Native Devops with Kubernetes: Building, Deploying, and Scaling Modern Applications in the Cloud*. Editorial O'Reilly Media

Sam Newman. (2021) *Building Microservices: Designing Fine-Grained Systems*. Editorial O'Reilly Media.

Mohamed Labouardy. (2022) *Pipeline as Code: Continuous Delivery with Jenkins, Kubernetes, and Terraform*. Manning Publications.

Rosemary Wang (2022) *Infrastructure as Code, Patterns and Practices: With Examples in Python and Terraform*. Manning Publications.