

1. Datos Generales de la asignatura

Nombre de la asignatura:	Desarrollo y Operaciones de Software (DevOps)
Clave de la asignatura:	IDB-2104
SATCA¹:	1 - 4 - 5
Carrera:	Ingeniería en Sistemas Computacionales

2. Presentación

Caracterización de la asignatura
<p>Esta asignatura le permitirá al alumno conocer y aplicar las herramientas tecnológicas necesarias para implementar una cultura de desarrollo de software de calidad. Es importante como desarrollador de software individual o como parte de un equipo de trabajo, integrarse de manera adecuada al proceso de desarrollo de aplicaciones de una manera eficiente para mejorar en sentido cuantitativo y cualitativo en la producción de software.</p> <p>En esta asignatura se verán herramientas de automatización dentro del proceso de desarrollo de software con el objetivo de entregar software de calidad, reduciendo tiempos y minimizando errores en dicho proceso.</p> <p>Esta materia se relaciona con todas las demás materias involucradas en el desarrollo de software como gestión de proyectos, ingeniería de software y programación web.</p>
Intención didáctica

¹ Sistema de Asignación y Transferencia de Créditos Académicos

Con esta asignatura se pretende proveer de las competencias necesarias para que el alumno pueda implementar el proceso y las herramientas relacionadas con la cultura de DevOps (*Development and Operations*).

En el primer tema se introduce a alumno al contexto que rodea a este modelo. Conociendo y aplicando los conceptos importantes para el desarrollo utilizando dicho modelo. A través del segundo tema, se comienza con la planeación del proyecto tomando como base una buena ingeniería de requerimientos y plasmándolos en una herramienta electrónica que permita monitorear el avance.

Con el tercer tema se utilizan herramientas de diseño, desarrollo y pruebas automatizadas para la creación y puesta a punto del software permitan una integración continua de los módulos desarrollados y mantenidos. Por ultimo, en el tema cuatro, y también a través de herramientas automatizadas se realiza el despliegue y monitoreo del funcionamiento de la aplicación desarrollada.

El docente deberá realizar practicas guiadas para la demostración de la implementación a manera de ejemplos, para posteriormente dejar al alumno una serie de ejercicios que le permitan reforzar lo aprendido en clase. Además, se sugiere indicar un proyecto de materia para aplicar los conocimientos en aplicaciones que resuelvan problemas de la vida real.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Jerez de García Salinas, Zacatecas. Mayo del 2021	Tecnológico Nacional de México Campus Jerez MTI Salvador Acevedo Sandoval	Reunión para el diseño y desarrollo de la especialidad para el periodo 2021 - 2023

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Aplicar las técnicas y herramientas de desarrollo de software que permitan mejorar la eficacia y eficiencia en la construcción y la entrega de productos de software.

5. Competencias previas

- Aplica metodologías e instrumentos, para garantizar la gestión adecuada de un proyecto de software.
- Desarrolla soluciones de software, considerando la metodología y herramientas para la elaboración de un proyecto aplicativo en diferentes escenarios.
- Conoce, identifica, selecciona y administra diferentes sistemas operativos con el fin de resolver problemáticas reales, así como aplicar procedimientos de interoperabilidad entre diferentes sistemas operativos.
- Configura y administra servicios de red para el uso eficiente y confiable de la infraestructura tecnológica de la organización.

6. Temario

1	Introducción a DEVOPS	1.1 Conceptos fundamentales 1.2 Herramientas de apoyo
2	Planificación	2.1 Comunicación con el equipo de trabajo 2.2 Planificación del proyecto
3	Desarrollo (CI)	3.1 Diseño y codificación 3.2 Integración continua del proyecto 3.3 Pruebas
4	Operaciones (CD)	4.1 Entrega 4.2 Despliegue continuo 4.3 Operaciones sobre el proyecto 4.4 Monitoreo

7. Actividades de aprendizaje de los temas

1. Introducción a DEVOPS	
Competencias	Actividades de aprendizaje
<p>Específica(s): Comprender los conceptos básicos para la aplicación de mejora de procesos.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de organizar y planificar. • Resolución de problemas. • Toma de decisiones. • Trabajo en equipo. • Capacidad de trabajar en un equipo interdisciplinar • Capacidad de aplicar los conocimientos en la práctica. • Diseño y gestión de proyectos. • Iniciativa y espíritu emprendedor. 	<ol style="list-style-type: none"> 1. Investigar los conceptos relevantes acerca de la mejora de proceso de desarrollo de software a través de DevOps. Elaborar un mapa conceptual. 2. Realizar una tabla comparativa analizando la forma tradicional de realizar software contra las nuevas tendencias como lo es DevOps.
2. Planificación	
Competencias	Actividades de aprendizaje
<p>Específica(s): Conocer y aplicar estrategias para la mejora de procesos en la fase de planificación del diseño y desarrollo de software a través de herramientas de automatización.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de organizar y planificar. • Resolución de problemas. • Toma de decisiones. • Trabajo en equipo. • Capacidad de trabajar en un equipo interdisciplinar • Capacidad de aplicar los conocimientos en la práctica. • Diseño y gestión de proyectos. • Iniciativa y espíritu emprendedor. 	<ol style="list-style-type: none"> 1. Investigar los conceptos importantes acerca de la fase de planificación del diseño y desarrollo de software. Elaborar un mapa conceptual. 2. Desarrollar ejercicios que permitan elaborar un plan de proyecto de software a través de herramientas automatizadas.
3. Desarrollo (CI)	
Competencias	Actividades de aprendizaje
<p>Específica(s): Conocer y aplicar estrategias para la mejora de procesos en la fase de desarrollo, pruebas e integración continua de software a través de herramientas de automatización.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de organizar y planificar. • Resolución de problemas. • Toma de decisiones. • Trabajo en equipo. 	<ol style="list-style-type: none"> 1. Investigar los conceptos importantes acerca de la fase de desarrollo, pruebas e integración continua. Elaborar un mapa conceptual. 2. Desarrollar ejercicios que permitan codificar, probar e integrar un proyecto de software a través de herramientas automatizadas.

<ul style="list-style-type: none"> • Capacidad de trabajar en un equipo interdisciplinar • Capacidad de aplicar los conocimientos en la práctica. • Diseño y gestión de proyectos. • Iniciativa y espíritu emprendedor. 	
<p>4. Operaciones (CD)</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s): Conocer y aplicar estrategias para la mejora de procesos en la fase operaciones sobre aplicaciones desplegadas, además de su monitoreo y control a través de herramientas de automatización.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de organizar y planificar. • Resolución de problemas. • Toma de decisiones. • Trabajo en equipo. • Capacidad de trabajar en un equipo interdisciplinar • Capacidad de aplicar los conocimientos en la práctica. • Diseño y gestión de proyectos. • Iniciativa y espíritu emprendedor. 	<ol style="list-style-type: none"> 1. Investigar los conceptos importantes acerca de la fase de operaciones sobre aplicaciones desplegadas. Elaborar un mapa conceptual. 2. Desarrollar ejercicios que permitan el despliegue y monitoreo continuo de un proyecto de software a través de herramientas automatizadas.

8. Práctica(s)

<ol style="list-style-type: none"> 1. Utilizar un software de administrador de proyectos para la ejecución y control de este. 2. Configurar un entorno para DevOps. 3. Utilizar software de control de versiones para codificar y llevar un control del repositorio. 4. Configurar un ambiente de pruebas. 5. Utilizar una herramienta para la integración continua. 6. Utilizar una herramienta para el despliegue continuo. 7. Elaborar aplicar un plan y procedimiento de pruebas con herramientas automatizadas. 8. Desplegar la aplicación hacia algún servicio web para generar un ambiente de producción. 9. Configurar una arquitectura de microservicios para el proyecto.
--

10. Integrar las practicas en un proyecto final.

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 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-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: resúmenes, mapas conceptuales, reportes de prácticas, estudios 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

1. Verona, J. (2016). Practical DevOps: Harness the power of DevOps to boost your skill set and make your IT organization perform better. Editorial PACKT
2. Guijarro Olivares, J. Caparrós Ramírez, J. y Cubero Luque, L. (2019). DevOps y seguridad cloud. Editorial UOC.
3. Jennifer Davis, Ryn Daniels (2016). Effective DevOps. Editorial O'Reilly Media, Inc.
4. Forsgren, N, Humble J., Kim G. (2018). Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations.
5. Gene Kim, Patrick Debois, John Willis, Jez Humble, John Allspaw. (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations.
6. by Joe Baron, Hisham Baz, Tim Bixler, Biff Gaut, Kevin E. Kelly, Sean Senior, John Stamper . AWS Certified Solutions Architect Official Study Guide: Associate Exam (Aws Certified Solutions Architect Official: Associate Exam).
7. Newma, S. Building Microservices: Designing Fine-Grained Systems 1st Edition
8. Soni, M. (2016). DevOps for Web Development. Editorial PACKT.
9. Farcic, V. The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Microservices
10. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective (SEI Series in Software Engineering) 1st Edition. Editorial Pearson