

DESARROLLO WEB FULLSTACK -INTERMEDIO

Autor de contenido

Andrés Fernando Pineda Guerra



Tabla de Contenido



Presentación

En el curso de desarrollador Full Stack como componente intermedio, podrán adquirir las habilidades y lenguajes necesarios para el desarrollo web, enfocándose en sus grandes pilares, como lo son Front End, Back End, Diseño y modelamiento de aplicaciones y documentación de código.

El curso trata temas emergentes tales como, la seguridad informática, desarrollo de aplicaciones móviles, gestión de base de datos, todo esto basado en la metodología Scrum. De la misma manera, se hace énfasis en el manejo de proyectos tanto en los módulos de desarrollo como los módulos de gestión de proyectos de TI.

Objetivos del curso (competencias)



Objetivo general

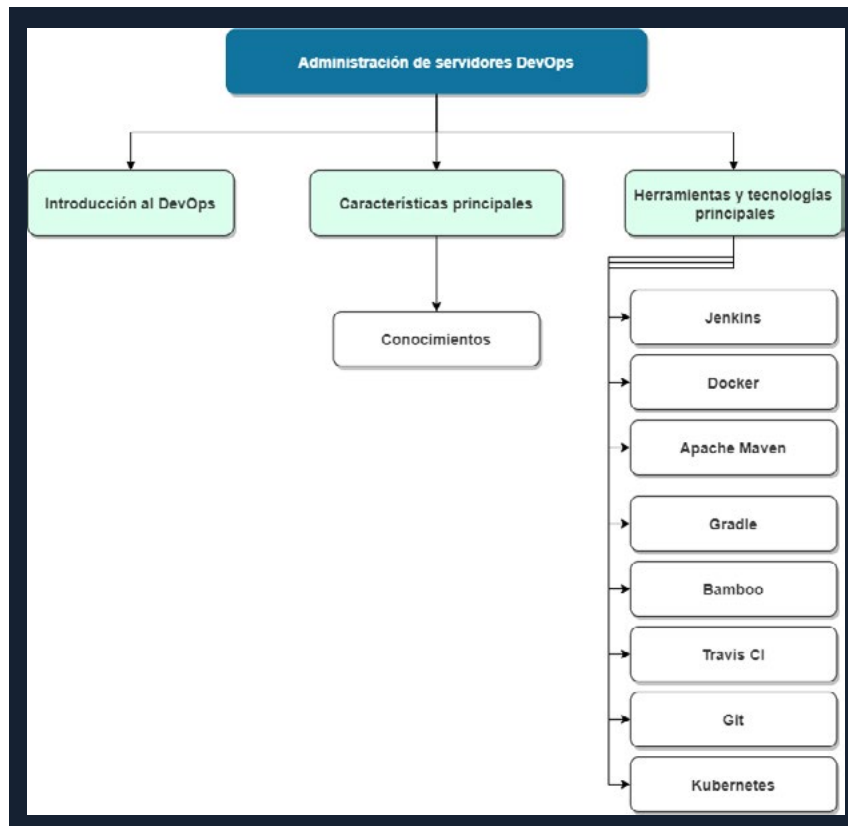
Formar a los participantes en el desarrollo web en todo el ciclo de vida del software, en donde adquieran los conocimientos básicos para implementar soluciones web.

Objetivo específico

- Conocer los conceptos y teoría básica del desarrollo web.
- Identificar y conocer los diferentes lenguajes de programación y herramientas para el desarrollo web.
- Aplicar las diferentes tecnologías web, tendencias y herramientas en el desarrollo de soluciones web enfocadas a proyectos.
- Diseñar, desarrollar e implementar soluciones web básicas en donde se integren los componentes de Front End, Back End, seguridad, redes y buenas prácticas utilizando metodologías ágiles.
- Identificar y conocer los conceptos básicos para el desarrollo móvil, así como aplicar su desarrollo en aplicaciones básicas.



Mapa de contenido de la unidad



Módulo 9 Administración de servidores DevOps

Ideas clave

Introducción a DevOps, principales características, conocimientos y habilidades. principales herramientas como Jenkins, Docker, Apache Maven, Gradle, Bamboo, Travis CI, Git y kubernetes.

9.1. Introducción al DevOps

DevOps es un modo de abordar la cultura, la automatización y el diseño de las plataformas para generar mayor valor empresarial y capacidad de respuesta, mediante la prestación ágil de servicios de alta calidad. Todo ello es posible gracias a la prestación rápida y constante de servicios de TI. DevOps implica vincular las aplicaciones heredadas con las aplicaciones creadas en la nube y las infraestructuras más nuevas.

El término “DevOps” es una combinación de las palabras “development” (desarrollo) y “operations” (operaciones), pero representa un conjunto de ideas y prácticas que van más allá de ambos conceptos, ya sea que estén juntos o separados. DevOps incluye sistemas de seguridad, maneras de trabajar en colaboración, análisis de datos, entre otras características.

DevOps describe los enfoques para agilizar los procesos con los que una idea (como una nueva función de software, una solicitud de mejora o una corrección de errores) pasa del desarrollo a la implementación, en un entorno de producción en que puede generar valor para el usuario. Estos enfoques requieren que los equipos de desarrollo y operaciones se comuniquen con frecuencia y aborden su trabajo con empatía hacia sus compañeros de equipo. También es necesario contar con capacidad de ajuste y tener flexibilidad a la hora de preparar los sistemas. Con DevOps, es posible brindar una mayor potencia a quienes más la necesitan mediante el autoservicio y la automatización. Los desarrolladores, que generalmente realizan codificaciones en un entorno de desarrollo estándar, trabajan en estrecha colaboración con los equipos de operaciones de TI para agilizar el diseño, las pruebas y el lanzamiento de los sistemas de software, sin comprometer la confiabilidad.

La finalidad de DevOps es crear las condiciones adecuadas para la colaboración entre los departamentos de desarrollo y de operaciones.

9.2. Características principales

- Los desarrolladores ya no están fuera de la gestión operativa. Con DevOps, los desarrolladores y administradores de sistemas se integran en un solo equipo de desarrollo (Dev) cohesivo y operativo (operaciones – Ops)
- DevOps nos empuja hacia un único código de referencia que incluye software, middleware y entorno operativo. DevOps reúne aquellos que históricamente fueron sistemas diferentes, en términos de modo de gestión, ciclo de lanzamiento o modelo de control. Con DevOps, más disciplinas trabajan juntas, en un modelo de entrega perfectamente integrado, para agregar valor al cliente.

- DevOps combina la integración continua (CI) y la entrega continua (CD), y el CI / CD mejora la velocidad para satisfacer las necesidades del cliente. ¡Es una cuestión de tiempo para comercializar!

Conocimientos:

- Los desarrolladores ya no están fuera de la gestión operativa. Con DevOps, los desarrolladores y administradores de sistemas se integran en un solo equipo de desarrollo (Dev) cohesivo y operativo (operaciones – Ops)
- DevOps nos empuja hacia un único código de referencia que incluye software, middleware y entorno operativo. DevOps reúne aquellos que históricamente fueron sistemas diferentes, en términos de modo de gestión, ciclo de lanzamiento o modelo de control. Con DevOps, más disciplinas trabajan juntas, en un modelo de entrega perfectamente integrado, para agregar valor al cliente.
- DevOps combina la integración continua (CI) y la entrega continua (CD), y el CI / CD mejora la velocidad para satisfacer las necesidades del cliente. ¡Es una cuestión de tiempo para comercializar!

Conocimientos:

- Experiencia con lenguajes de scripting: Para adentrarnos en el mundo DevOps es muy importante tener una base sólida de programación aún sin ser desarrolladores expertos. Los lenguajes más relevantes a día de hoy para tareas relacionadas con DevOps son Bash y Python, aunque desde Geko recomendamos tener conocimientos de algún lenguaje adicional como por ejemplo: Go, C, C++, PHP, Ruby, Node.js o PowerShell. Aunque no vayamos a programar sí necesitaremos conocer las tecnologías con las que trabajan los desarrolladores para poder realizar la implementación de las pipelines de integración, así como desarrollar nuestras propias herramientas para la automatización de algunos procesos.
- Security Skills: Este es uno de los puntos que añaden más valor como profesional DevOps. Los ingenieros deben tener las habilidades para escribir código seguro para proteger las aplicaciones de ataques no deseados. Unos tiempos de desarrollo más rápidos conllevan un mayor riesgo de que se introduzcan más vulnerabilidades en el código.
- Contenedores y orquestación de contenedores: Muchas empresas abandonan los modelos tradicionales de despliegue de aplicaciones para realizar una transición a un entorno basado en microservicios (generalmente sistemas en contenedores). Las ventajas del uso de contenedores son múltiples, desde asegurar que lo mismo

que se prueba en desarrollo funcionará exactamente igual en todos los entornos, hasta automatizar más fácilmente el despliegue del código en producción. Por ello, es importante conocer las herramientas para la administración y gestión de contenedores, algunas de estas herramientas incluyen: Docker, Kubernetes, Docker Swarm, ECS y muchas más.

- **Servicios web :** Los servidores web se utilizan para satisfacer las solicitudes de contenidos de un sitio web por parte de los usuarios. Si estás interesado en el mundo DevOps seguramente hayas escuchado que existen diferentes tipos de servidores web, por ello os mencionamos los más conocidos: Nginx, Apache, IIS y Tomcat.
- **Automatización de configuración como código :** Ya sea para aprovisionar aplicaciones y sus configuraciones, crear usuarios, recursos, automatizar las tareas propias de la administración de sistemas; las herramientas de automatización de configuración permiten mantener un estado consistente a lo largo del tiempo y de la infraestructura gestionada. Algunas de las herramientas más populares para esta automatización son Ansible, Chef, Puppet y Salt.
- **Cloud Computing:** La nube y los DevOps generalmente van juntos de la mano. De un ingeniero DevOps se espera que sepa manejar las distintas plataformas cloud y las herramientas de las mismas para adaptarse mejor a las necesidades, y buscar soluciones para cada cliente o proyecto. Además, también deberá conocer las diferentes plataformas en la nube para sopesar los pros y los contras de utilizar servicios IaaS frente a PaaS, y otros servicios gestionados para diseñar y recomendar soluciones. Plataformas como AWS, Azure, Google Cloud Platform, etc deben ser herramientas conocidas por un ingeniero DevOps.
- **Implementación de infraestructura como código:** Independientemente del entorno o del sistema operativo, un buen DevOps debe tener conocimientos sólidos en la administración de infraestructura y sistemas operativos, y ser capaz de automatizar el despliegue y configuración de la misma usando herramientas de infraestructura como código. Las herramientas más relevantes en este sector son Terraform y CloudFormation, aunque esta última solo sirve para AWS.
- **Implementación de despliegues CI/CD:** Para este puesto necesitarás un profundo conocimiento de la esencia de DevOps: la implementación de despliegues, conocido como CI/CD. Todo el código se fusiona y, finalmente, se ejecuta la construcción y las pruebas se automatizan. El despliegue a producción requerirá sobre todo la creación de nuevos contenedores, automatizando los procesos mediante scripts. Algunas herramientas que debe conocer son: Gitlab CI, GitHub Actions, Bamboo, Azure DevOps, Jenkins, Travis CI, TeamCity y Circle CI

- **Monitorización de la infraestructura:** Una de las partes fundamentales del trabajo de un DevOps es la de saber qué ocurre en todo momento en la aplicación una vez finaliza su etapa de desarrollo. Por ello, monitorizar la infraestructura es una tarea esencial, ya que permite visualizar los eventos y recibir alertas en tiempo real. Algunas de las herramientas DevOps para la monitorización, que ayudan a los equipos de desarrollo y operaciones a trabajar juntos de forma eficaz, son: Prometheus, Grafana, Sensu, y New Relic Infrastructure.
- **Metodologías ágiles:** Un ingeniero DevOps debe dominar las metodologías ágiles que le permitan realizar un desarrollo iterativo e incremental. Esta metodología tiene como eje central el desarrollo iterativo con enfoque incremental y evolutivo. Esto quiere decir que el desarrollo del software es dividido o fragmentado en sprints o tareas más pequeñas. Por ello, existen metodologías como Kanban o Scrum, y herramientas que las ofrecen o soportan, como Jira.
- **Experiencia con lenguajes de scripting:** Para adentrarnos en el mundo DevOps es muy importante tener una base sólida de programación aún sin ser desarrolladores expertos. Los lenguajes más relevantes a día de hoy para tareas relacionadas con DevOps son Bash y Python, aunque desde Geko recomendamos tener conocimientos de algún lenguaje adicional como por ejemplo: Go, C, C++, PHP, Ruby, Node.js o PowerShell. Aunque no vayamos a programar sí necesitaremos conocer las tecnologías con las que trabajan los desarrolladores para poder realizar la implementación de las pipelines de integración, así como desarrollar nuestras propias herramientas para la automatización de algunos procesos.
- **Security Skills:** Este es uno de los puntos que añaden más valor como profesional DevOps. Los ingenieros deben tener las habilidades para escribir código seguro para proteger las aplicaciones de ataques no deseados. Unos tiempos de desarrollo más rápidos conllevan un mayor riesgo de que se introduzcan más vulnerabilidades en el código.
- **Contenedores y orquestación de contenedores:** Muchas empresas abandonan los modelos tradicionales de despliegue de aplicaciones para realizar una transición a un entorno basado en microservicios (generalmente sistemas en contenedores). Las ventajas del uso de contenedores son múltiples, desde asegurar que lo mismo que se prueba en desarrollo funcionará exactamente igual en todos los entornos, hasta automatizar más fácilmente el despliegue del código en producción. Por ello, es importante conocer las herramientas para la administración y gestión de contenedores, algunas de estas herramientas incluyen: Docker, Kubernetes, Docker Swarm, ECS y muchas más.

- **Servicios web :** Los servidores web se utilizan para satisfacer las solicitudes de contenidos de un sitio web por parte de los usuarios. Si estás interesado en el mundo DevOps seguramente hayas escuchado que existen diferentes tipos de servidores web, por ello os mencionamos los más conocidos: Nginx, Apache, IIS y Tomcat.
- **Automatización de configuración como código :** Ya sea para aprovisionar aplicaciones y sus configuraciones, crear usuarios, recursos, automatizar las tareas propias de la administración de sistemas; las herramientas de automatización de configuración permiten mantener un estado consistente a lo largo del tiempo y de la infraestructura gestionada. Algunas de las herramientas más populares para esta automatización son Ansible, Chef, Puppet y Salt.
- **Cloud Computing:** La nube y los DevOps generalmente van juntos de la mano. De un ingeniero DevOps se espera que sepa manejar las distintas plataformas cloud y las herramientas de las mismas para adaptarse mejor a las necesidades, y buscar soluciones para cada cliente o proyecto. Además, también deberá conocer las diferentes plataformas en la nube para sopesar los pros y los contras de utilizar servicios IaaS frente a PaaS, y otros servicios gestionados para diseñar y recomendar soluciones. Plataformas como AWS, Azure, Google Cloud Platform, etc deben ser herramientas conocidas por un ingeniero DevOps.
- **Implementación de infraestructura como código:** Independientemente del entorno o del sistema operativo, un buen DevOps debe tener conocimientos sólidos en la administración de infraestructura y sistemas operativos, y ser capaz de automatizar el despliegue y configuración de la misma usando herramientas de infraestructura como código. Las herramientas más relevantes en este sector son Terraform y CloudFormation, aunque esta última solo sirve para AWS.
- **Implementación de despliegues CI/CD:** Para este puesto necesitarás un profundo conocimiento de la esencia de DevOps: la implementación de despliegues, conocido como CI/CD. Todo el código se fusiona y, finalmente, se ejecuta la construcción y las pruebas se automatizan. El despliegue a producción requerirá sobre todo la creación de nuevos contenedores, automatizando los procesos mediante scripts. Algunas herramientas que debe conocer son: Gitlab CI, GitHub Actions, Bamboo, Azure DevOps, Jenkins, Travis CI, TeamCity y Circle CI
- **Monitorización de la infraestructura:** Una de las partes fundamentales del trabajo de un DevOps es la de saber qué ocurre en todo momento en la aplicación una vez finaliza su etapa de desarrollo. Por ello, monitorizar la infraestructura es una tarea esencial, ya que permite visualizar los eventos y recibir alertas en tiempo real. Algunas de las herramientas DevOps para la monitorización, que ayudan a los equipos de desarrollo y operaciones a trabajar juntos de forma eficaz, son: Prometheus, Grafana, Sensu, y New Relic Infrastructure.

- Metodologías ágiles: Un ingeniero DevOps debe dominar las metodologías ágiles que le permitan realizar un desarrollo iterativo e incremental. Esta metodología tiene como eje central el desarrollo iterativo con enfoque incremental y evolutivo. Esto quiere decir que el desarrollo del software es dividido o fragmentado en sprints o tareas más pequeñas. Por ello, existen metodologías como Kanban o Scrum, y herramientas que las ofrecen o soportan, como Jira.

9.3. Herramientas y tecnologías principales

Jenkins



Jenkins

Jenkins es un servidor de automatización de código abierto y gratuito que ayuda a automatizar los procesos de desarrollo de software, como la construcción, facilitando el CI/CD, el lanzamiento y los testeos.

Esta herramienta DevOps facilita a los equipos la supervisión de las tareas repetidas, la integración de los cambios con facilidad y la identificación de los problemas con rapidez.

Docker



Líder en la contenedorización de software, Docker es utilizado por más de 11 millones de desarrolladores en todo el mundo. Solomon Hykes es su autor original y fue lanzado en 2013 por Docker, Inc.

Como herramienta de DevOps, Docker ayuda a los desarrolladores a construir, empaquetar y luego desplegar los códigos con facilidad y rapidez a través de contenedores con las dependencias necesarias en lugar de máquinas virtuales. Elimina las actividades mundanas de configuración y fomenta la colaboración eficaz del equipo.

Apache Maven



Ayuda a gestionar la construcción, la documentación y los informes de un proyecto. Además de los proyectos basados en Java, también puedes utilizar Maven para desarrollar y gestionar proyectos de software escritos en Ruby, C#, Scala y otros.

Gradle



Acelera la productividad de tu desarrollo de software utilizando Gradle. También se encuentra entre las herramientas DevOps de código abierto para la automatización de la construcción, especialmente para el desarrollo de aplicaciones o software en varios idiomas.

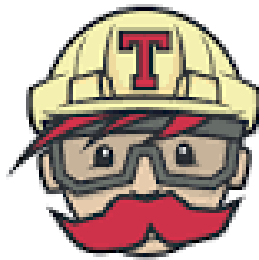
Bamboo



Bamboo

Ayuda a crear planes de construcción en múltiples etapas, a establecer disparadores y a asignar agentes a las construcciones críticas, así como a los despliegues

Travis CI



Travis CI

Más adecuado para proyectos de código abierto, Travis CI es una eficiente herramienta de integración continua (CI) para desarrollar y probar aplicaciones alojadas en GitHub o Bitbucket. Está escrito en Ruby y tiene una licencia MIT.

Este servicio alojado en la nube detecta todos los nuevos commits de código automáticamente y cuántos códigos se empujan a su repositorio de GitHub para ayudarte a construir el software y probarlo en consecuencia.

Git



git

Esta herramienta de sistemas de control de código fuente (SCM – Source code management) distribuida se utiliza para hacer un seguimiento de los cambios en los archivos para el desarrollo de software y coordinar el trabajo de forma eficaz entre los programadores. Su objetivo es aumentar la velocidad, la compatibilidad y la integridad de los datos en los flujos de trabajo distribuidos no lineales.

Kubernetes



kubernetes

De las mejores herramientas de automatización de DevOps, Kubernetes es útil para aplicaciones empaquetadas con muchos contenedores. Permite a los desarrolladores aislar y empaquetar clústeres de contenedores de forma lógica para facilitar el despliegue en varias máquinas.

Otros materiales para profundizar

Recursos de video



Cultura DevOps (Director). (2021, febrero 1). ¿Que necesitas saber para ser DEVOPS? Ruta para ser un DEVOPS. <https://www.youtube.com/watch?v=ZosASq10X-HQ>

EDteam (Director). (2022). ¿Qué es y qué no es DevOps? <https://www.youtube.com/watch?v=MtDFK-evWw4>

Referencias bibliográficas de la unidad



A Survey of DevOps Concepts and Challenges | ACM Computing Surveys. (s. f.). Recuperado 3 de enero de 2023, de <https://dl.acm.org/doi/abs/10.1145/3359981>

Díaz Cortés, E. A. (2018). Implantación de procesos y herramientas basadas en DevOps para una empresa de servicios de información. <https://repositorio.uchile.cl/handle/2250/169243>

Zhu, L., Bass, L., & Champlin-Scharff, G. (2016). DevOps and Its Practices. IEEE Software, 33(3), 32-34. <https://doi.org/10.1109/MS.2016.81>



**ALCALDÍA MAYOR
DE BOGOTÁ D.C.**
SECRETARÍA DE EDUCACIÓN



ATENEA
AGENCIA DISTRITAL PARA LA EDUCACIÓN
SUPERIOR LA CIENCIA Y LA TECNOLOGÍA



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**
Acreditación Institucional de Alta Calidad