
Universidad
Tecmilenio®

Universidad Tecmilenio

Operaciones y Desarrollo: DevOps

Arquitectura de microservicios. Parte 1

Semana 1

Introducción

En el mundo de DevOps, la arquitectura de microservicios se está convirtiendo en un marco bastante común para construir servicios. Cuando se trata de microservicios, la arquitectura suele desglosarse en pequeños componentes basados en servicios diseñados para resolver una necesidad empresarial específica. Estos componentes tienen funcionalidades específicas como el registro y la monitorización.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

Una arquitectura de **microservicios** es una colección de servicios autónomos, pequeños y de propósito definido. Cada servicio es independiente y debe implementar una funcionalidad de negocio específica dentro de un contexto delimitado.

En una arquitectura de microservicios, cada servicio se desarrolla, despliega, opera y escala de **forma independiente**, sin necesidad de compartir código, marcos de desarrollo, lenguajes, implementaciones o repositorios de servicios con otros servicios.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

Ventaja del uso de los microservicios:

- Agilidad.
- Equipos pequeños y centrados.
- Base de código pequeña.
- Mezcla de tecnologías.
- Aislamiento de errores.
- Escalabilidad.
- Aislamiento de datos.

Desafíos en el uso de los microservicios:

- Complejidad.
- Desarrollo y pruebas.
- Falta de estandarización.
- Congestión y latencia de red.
- Integridad de datos.
- Administración.
- Control de versiones.

Fuente: Ilimit. (2020). *Arquitecturas monolíticas o arquitectura de microservicios: ventajas e inconvenientes*. Recuperado de https://www.ilimit.com/blog/arquitecturas-monoliticas-o-arquitectura-de-microservicios-ventajas-e-inconvenientes/

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

Tipos de arquitectura en microservicios

Monolítica

- Tienden a utilizar un único código base para sus distintos servicios y funciones. Esto significa que la aplicación tiene un control total sobre todas las tareas necesarias para llevar a cabo una determinada función.

Distribuida

- Consiste en tener la aplicación desagregada en múltiples partes. Esta arquitectura es mucho más compleja y difícil de gestionar. Sin embargo, ofrece la ventaja del escalado horizontal y el manejo de elasticidad.

SOA

- Se centra más en las aplicaciones, mientras que Cloud Native se centra más en la arquitectura del sistema de alojamiento y en el uso de los recursos de la nube bajo demanda. Cloud Native tiene más elasticidad, que es gestionada automáticamente por la infraestructura dedicada a las aplicaciones que la distribuyen.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

Modelado de servicios

Analizar el dominio

Es necesario conocer los requisitos funcionales de la aplicación para crear un conjunto más formal de modelos de dominio. De este modo, podrás obtener una descripción más detallada del dominio.

Definir contextos

Cada contexto delimitado presenta su propio modelo de dominio que representa un subdominio específico dentro de la aplicación más amplia.

Modelos tácticos

Para definir las entidades de dominio, los agregados y los servicios para cada contexto delimitado se aplican los modelos de diseño basados en el dominio táctico. Los agregados suelen ser colecciones de objetos de valor o entidades que sirven a una entidad raíz.

Identificar microservicios

En este paso se definirán las:

Entidades.

Objetos de valor.

Agregados.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

Partiendo del monolito

Al migrar de una arquitectura monolítica a una de microservicios, es crucial evaluar el entorno por adelantado. Esto se debe a que, al dividir el monolito en microservicios más pequeños, existe el reto potencial de definir el alcance y las limitaciones (Ilimit, 2020).

Fortalezas de la arquitectura monolítica

1. Menos puntos de quiebre.
2. Facilidad para depurar (*debugging*).
3. Despliegue sencillo.
4. Facilidad para desarrollar.

Debilidades de la arquitectura monolítica

1. Entendimiento.
2. Dificultad en cambios de infraestructura.
3. Susceptibilidad a fallas.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Ejercicio

Una empresa que se dedica a la mensajería exprés te pide que desarrolles una solución basada en la nube para coordinar su servicio de entregas, ya que han tenido un crecimiento acelerado en el último año. Están teniendo un crecimiento tal, que requieren duplicar su flotilla cada mes y están subcontratando a personas que tengan vehículos de carga cerrados, previa inspección del vehículo, prueba de manejo y participación en un curso de capacitación en línea (pasando un examen final) sobre servicio al cliente. El servicio de mensajería exprés funciona de la siguiente manera: una persona o negocio solicita un viaje donde se recogen paquetes desde una dirección de origen y se llevan a una dirección destino antes de que transcurran 24 horas. El servicio está restringido al área metropolitana. El cobro al usuario final se hace con base en kilómetro recorrido, donde se cobra una tarifa fija que se reparte un porcentaje al subcontratista y un porcentaje para la empresa que provee la plataforma.

Utiliza la metodología de diseño determinado por dominios (DDD) como base para diseñar los microservicios necesarios para el caso descrito con anterioridad.

1) Define e identifica las entidades que participan en el proceso. Asegúrate de que las entidades tengan un contexto definido, atributos y métodos (acciones que pueden afectar a la entidad). Recuerda que solo la entidad puede acceder y modificar sus datos y que otras entidades deben acceder a ellos a través de invocaciones a los métodos a través de un API (*application program interface*). Recuerda que las entidades tienen identidad única y pueden ser invocadas en otros microservicios.
2) Define e identifica los objetos de valor a ser incluidos en el proceso. Recuerda que los objetos de valor no tienen identidad propia y solo tienen sentido dentro del contexto de un microservicio.
3) Define e identifica los agregados en el modelo. Recuerda que los agregados son colecciones que aglutinan varias entidades alrededor de una entidad raíz.
4) El documento debe incluir una justificación del motivo para utilizar microservicios sobre una aplicación monolítica.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Cierre

Es importante distinguir cuando una arquitectura se debe basar en microservicios, para ello, debes considerar los siguientes puntos para implementar una arquitectura basada en microservicios:

¿Cuándo será un buen momento para migrar a esta arquitectura?
¿Qué reglas /lineamientos se pueden seguir para identificar las fronteras de microservicios utilizando técnicas como DDD (*domain driven design*)?
¿Cómo se puede diseñar una buena manera para que los servicios colaboren entre sí, sin incrementar el acoplamiento entre ellos, usando colaboración basada en eventos?

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Bibliografía

Ilimit. (2020). *Arquitecturas monolíticas o arquitectura de microservicios: ventajas e inconvenientes*. Recuperado de https://www.ilimit.com/blog/arquitecturas-monoliticas-o-arquitectura-de-microservicios-ventajas-e-inconvenientes/

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Universidad Tecmilenio

Operaciones y Desarrollo: DevOps

Arquitectura de microservicios. Parte 2

Semana 1

Introducción

En la era moderna, el software se entrega comúnmente como un servicio. La metodología de 12 factores crea aplicaciones como servicio. minimizando la divergencia entre desarrollo y producción. Utiliza formatos declarativos para la automatización, se implementa en la nube y permite la escalabilidad y portabilidad entre entornos de ejecución.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

Metodología de los 12 factores (Reselman, 2021).

1

Repositorio central

- Se usa para dar seguimiento de una base de código en el control de revisiones, en donde parte una gran cantidad de despliegues.
Una aplicación de 12 factores siempre se rastrea en un sistema de control de versiones como Git, Mercurial o Subversion. Una copia de la base de datos de seguimiento de revisiones se conoce como repositorio de código, a menudo abreviado como repositorio de código o simplemente repositorio.

2

Dependencias

- Una aplicación de 12 factores nunca se basa en la existencia implícita de paquetes en todo el sistema. Declara todas las dependencias, completa y exactamente a través de un manifiesto de declaración de dependencia.

3

Configuración

- La configuración de una aplicación es todo lo que puede variar entre implementaciones (*staging*, producción, entornos de desarrollo, etc.). Esto incluye:
  - Controladores de recursos para la base de datos, Memcached y otros servicios de respaldo.
  - Credenciales para servicios externos como Amazon S3 o Twitter.
  - Valores por implementación, como el nombre del host.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

Metodología de los 12 factores

4

Servicios de soporte

- Un servicio de respaldo es cualquier servicio que la aplicación consume a través de la red como parte de su funcionamiento normal. Los ejemplos incluyen almacenes de datos, sistemas de mensajería/colas, servicios SMTP para correo electrónico saliente y sistemas de almacenamiento en caché.

5

Construir, desplegar, ejecutar

- La **etapa de construir** es una transformación que convierte un repositorio de código en un paquete ejecutable conocido como compilación.
- La **etapa de lanzamiento** toma la compilación producida y la combina con la configuración actual de la implementación. La versión resultante está lista para el entorno de ejecución.
- La **etapa de ejecución** implementa la aplicación en el entorno de ejecución, mediante el lanzamiento de un conjunto de procesos de la aplicación.

6

Procesos

- La aplicación se ejecuta en el entorno de ejecución como uno o más procesos.
Los procesos de 12 factores son huérfanos y no comparten nada. Cualquier dato que deba persistir debe almacenarse en un servicio de respaldo con estado, generalmente una base de datos.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

Metodología de los 12 factores

7 Asignación de puertos

- En este factor se realiza la exportación de servicios a través de la vinculación de puertos en la red.
- Las aplicaciones web a veces se ejecutan dentro de un contenedor de servidor web.

8 Concurrencia

- Las aplicaciones web han adoptado una variedad de formas de ejecución de procesos.
- En la aplicación de 12 factores, los procesos son ciudadanos de primera clase.
- Con este modelo, el desarrollador puede diseñar su aplicación para manejar diversas cargas de trabajo.

9 Prescindibilidad

- Los procesos de la aplicación de 12 factores son prescindibles, lo que significa que se pueden iniciar o detener en cualquier momento. Esto facilita el escalado elástico rápido, la implementación rápida de cambios de código o configuración y la solidez de las implementaciones de producción.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

Metodología de los 12 factores

10

Paridad Dev/Prod

- El desarrollo, la puesta en escena y la producción tienen que ser lo más similares posible; esto es para disminuir los errores en la codificación que afecten los tiempos de despliegue de las aplicaciones.
- Históricamente, ha habido brechas sustanciales entre el desarrollo y la producción.
- La aplicación de 12 factores está diseñada para una implementación continua al mantener pequeña la brecha entre el desarrollo y la producción.

11

Bitácoras (logs)

- Las bitácoras son un historial de eventos para llevar un control más estricto, así como una supervisión en los comportamientos de las aplicaciones.
- Las bitácoras proporcionan visibilidad del comportamiento de una aplicación en ejecución. En entornos basados en servidor, normalmente se escriben en un archivo en el disco (un "archivo de bitácora" o "*logfile*"), pero este es solo un formato de salida.

12

Procesos de administración

- Las bitácoras son un historial de eventos para llevar un control más estricto, así como una supervisión en los comportamientos de las aplicaciones.
- Las bitácoras proporcionan visibilidad del comportamiento de una aplicación en ejecución. En entornos basados en servidor, normalmente se escriben en un archivo en el disco (un "archivo de bitácora" o "*logfile*"), pero este es solo un formato de salida.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Ejercicio

Realiza lo siguiente:

Escribe un ensayo sobre la forma en la que utilizarías la metodología de apps de 12 factores para construir aplicaciones de software como servicio (SaaS, por sus siglas en inglés).

La longitud del ensayo debe ser de al menos tres cuartillas.
Incluye cómo utilizarías cada uno de los 12 factores en el escenario SaaS: repositorio central, dependencias, respaldos, configuración, compilación y ejecución de software, procesos, asignación de puertos, concurrencia, prescindibilidad, paridad, bitácoras y procesos de administración.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Cierre

Es importante que sepas distinguir cuándo una arquitectura de microservicios es aplicable para resolver los problemas de escalamiento que pudiera tener tu aplicación. Si tu equipo es pequeño o tu aplicación aún no alcanza un volumen de usuarios o transaccional de gran escala, quizás aún haya tiempo de planear y refinar el diseño antes de migrar a una estructura que implica mayor disciplina y procesos de administración de clase mundial.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Bibliografía

Reselman, B. (2021). *An illustrated guide to 12 Factor Apps*. Recuperado de https://www.redhat.com/architect/12-factor-app

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Universidad Tecmilenio

Operaciones y Desarrollo: DevOps

Introducción a DevOps

Semana 1

Introducción

Las organizaciones que se basan en la filosofía DevOps disfrutan de un desarrollo de software acelerado, con procesos de mayor calidad y más fiables. Esta filosofía consta de varias etapas, como el desarrollo continuo, la integración continua y las pruebas continuas.

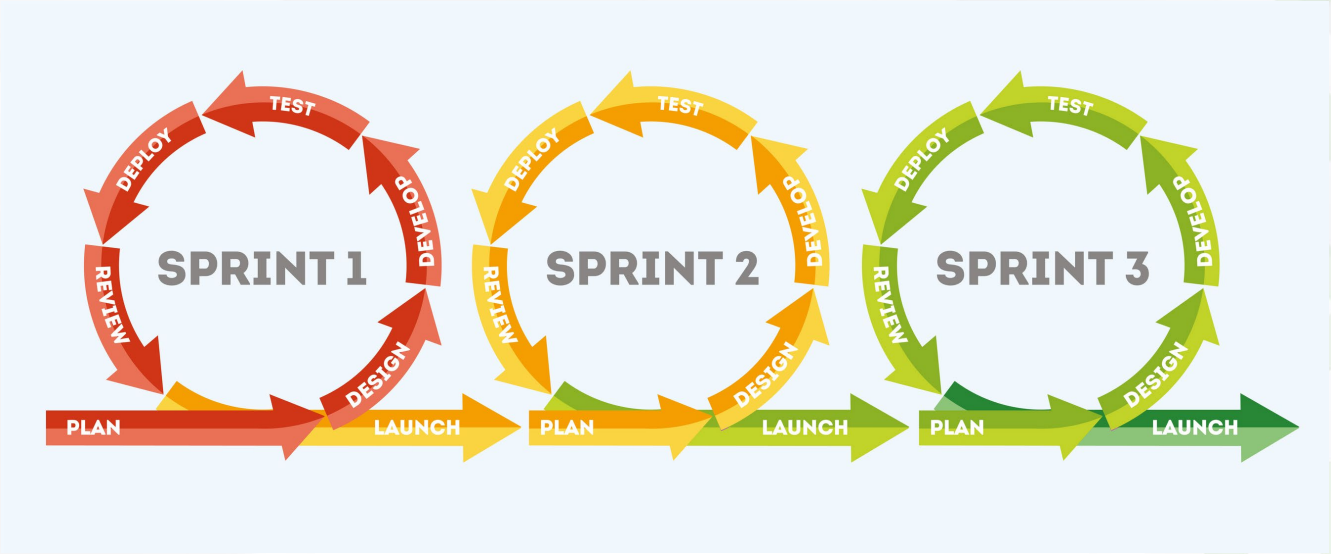
DevOps es una combinación de dos palabras: desarrollo y operaciones.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

En DevOps se utilizan prácticas de la metodología ágil, ya que el enfoque es el desarrollo de software repetitivo o iterativo en el que el proyecto de software se divide en *sprints*.

Cada sprint tiene distintas fases, como el análisis y recopilación de requerimientos, el desarrollo del diseño de software, las pruebas funcionales y no funcionales del sistema, y el mantenimiento. Estos sprints son utilizados en DevOps (Microsoft, s.f.).



Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

Conceptos básicos de DevOps

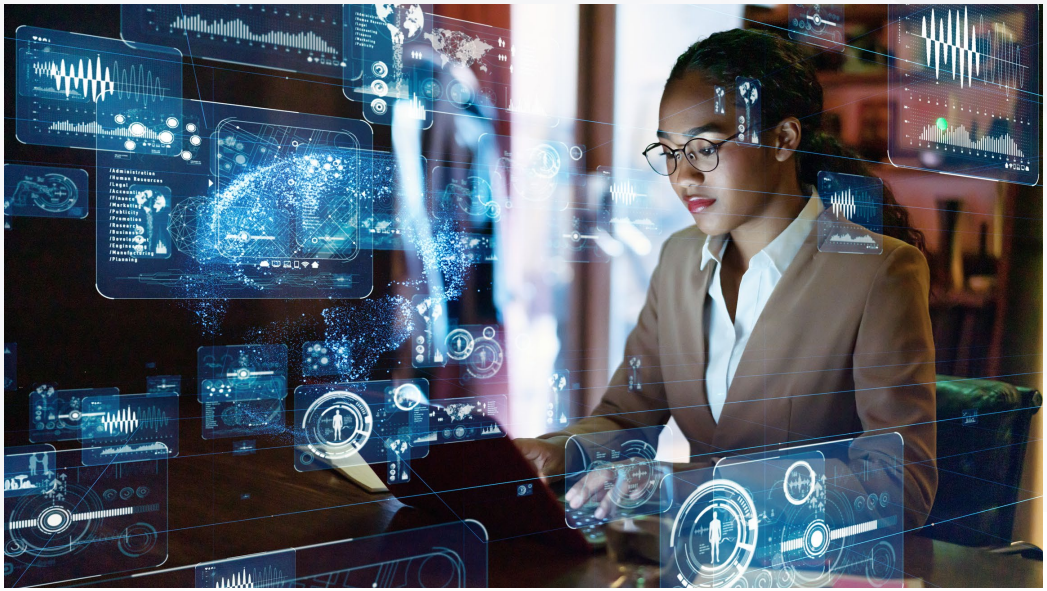
1. **Desarrollo continuo:** se enfoca en construir el código actual, sin tiempo de espera para la implementación.
2. **Integración continua (CI, continuous integration):** mecanismo de retroalimentación, desarrollo del código y pruebas más rápidas.
3. **Entregas continuas:** se preparan automáticamente los cambios en el código para producción.
4. **Despliegues continuos (CD, continuous deployment):** se implementa el código en los servidores de producción.
5. **Monitoreo continuo:** se requiere un buen sistema de retroalimentación y monitoreo.
6. **Infraestructura como código (IaC, infrastructure as code):** gestión de las redes, máquinas virtuales, balanceadores de carga y topología de conexión en un modelo descriptivo utilizando código fuente
(Microsoft, 2021).

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

En la fase del **desarrollo continuo** se lleva a cabo la planificación y la codificación del software. Basándose en los requerimientos se define la planificación y el alcance del proyecto, así los desarrolladores empiezan a construir el código para la aplicación

En la fase de **pruebas continuas** se detectan los errores del software desarrollado, ya que se realizan las pruebas continuamente. En la fase se utilizan herramientas para automatización de pruebas. Estas permiten que los QA (*quality assurance* o control de calidad) realicen múltiples pruebas de la codificación para revisar con cuidado y evitar fallas en la funcionalidad.



Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

La **fase del monitoreo continuo:** en esta etapa se realiza la supervisión continua del rendimiento del sistema o aplicación y es crucial en el ciclo de vida.

La fase de **integración continua** se puede considerar como la más importante en el ciclo de vida de DevOps, es en donde se realizan los cambios en el código fuente con más frecuencia. **El flujo de trabajo** no solo sería la compilación del código, sino que, además, se tiene que revisar, probar, integrar las pruebas y, por último, empaquetar.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Explicación

Casos de éxito

Netflix

Su proceso de ingeniería de software muestra un entendimiento profundo de la metodología que se enfoca en atributos de calidad, a través de procesos automatizados.

Su servicio de transmisión de video es un gran sistema distribuido que se aloja en Amazon Web Services y se integra de cientos de microservicios.

Target

Implemento la filosofía DevOps a partir de un cambio cultural dentro de la organización. Según el CEO, Mike McNamara, una parte esencial de la estrategia de Target fue la creación de un periodo de trabajo de seis semanas llamado Dojo, en el que los equipos realizaron su trabajo junto a entrenadores ágiles que los apoyaban mientras interactuaban con DevOps.

Walmart

En 2011, Walmart Labs lanzaron su brazo de innovación y desarrollo tecnológico (King, 2018). El objetivo fue adoptar DevOps y garantizar entregas continuas. Además, incorporaron OneOps, una tecnología en la nube que automatiza y acelera el proceso de despliegue de aplicaciones de cómputo. Asimismo, crearon diferentes herramientas de código abierto que permiten construir aplicaciones y servicios a partir de código reutilizable.

Amazon

Migró su infraestructura a Amazon Web Services (AWS) después de experimentar problemas al dimensionar la cantidad de servidores dedicados. Esta migración ha facilitado a sus ingenieros ajustar la capacidad de cómputo de forma incremental. Esto redujo el gasto en infraestructura y estimuló una transición a un proceso de implementación continuo.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Ejercicio

1. En un documento de Word crea una tabla comparativa por cada etapa, en la cual evaluarás los atributos de las herramientas que actualmente se usan en las fases de desarrollo, pruebas, integración, despliegue y monitoreo continuo. La tabla es así:

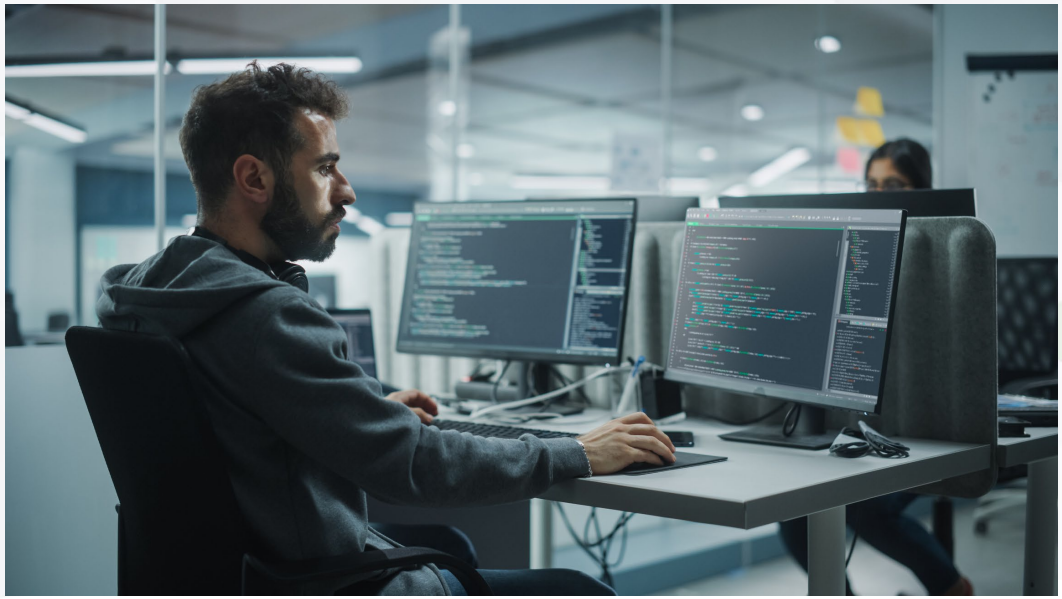
Atributo	Herramienta 1	Herramienta 2	Herramienta 3
Sistemas operativos en los que opera			
Costo			
Entrenamiento disponible			
Soporte técnico			
Empresas conocidas que los usan			
Lista de funcionalidades escenciales			
Plugins			
Integraciones			
Otros/comentarios			

2. Cada tabla debe incluir cuál es tu recomendación de la herramienta a utilizar y la justificación.

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

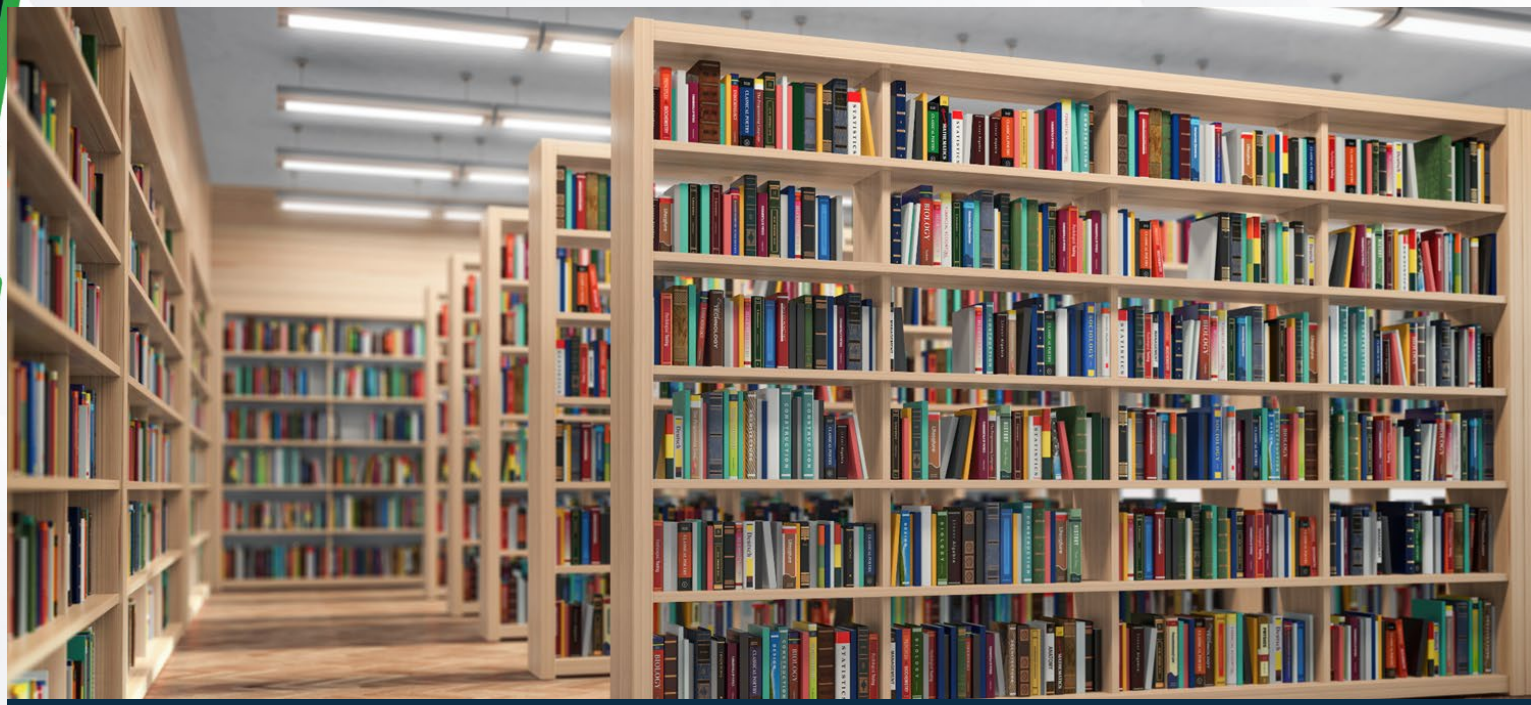
Cierre

Además de los esfuerzos por romper las barreras de comunicación y fomentar la colaboración entre los equipos de desarrollo y operaciones tecnológicas, uno de los principales valores de DevOps es lograr la satisfacción del cliente y prestar sus servicios en menos tiempo.
DevOps también se ha creado para impulsar la innovación empresarial y ser el motor de mejoras continuas en los procesos.
La práctica de DevOps propicia que cada empresa se ponga como objetivo ofrecer un mejor servicio, en menos tiempo, de mejor calidad y con mayor seguridad a sus clientes finales.



Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .

Bibliografía



King, J. (2018). *Walmart Labs: Our Journey to Change the Way the World Shops*. Recuperado de https://medium.com/walmartglobaltech/walmart-labs-our-journey-to-change-the-way-the-world-shops-788a1908cf92

Microsoft. (s.f.). *¿Qué es DevOps?* Recuperado de https://azure.microsoft.com/es-mx/overview/what-is-devops/#devops-overview

Microsoft. (2021). *What is Infrastructure as Code?* Recuperado de https://docs.microsoft.com/en-us/devops/deliver/what-is-infrastructure-as-code

Derechos de Autor Reservados. (UNIVERSIDAD TECMILENIO®) .