



Universidad
Tecnológica
de Bolívar

DevOps

Presentado por:
Elías José Blanco
Juan Felipe Silgado
Mateo De Jesús Reyes
Miguel Andrés Villa
Sebastián Valencia

DevOps

- Se trata de colaboración. Los equipos de desarrollo y operaciones se fusionan en un equipo funcional que se comunica, comparte feedback y colabora durante todo el ciclo de desarrollo e implementación.
- Se responsabilizan de una función o un proyecto a lo largo del ciclo de vida completo, desde la concepción hasta la entrega. Este nivel superior de dedicación y compromiso del equipo da lugar a unos resultados de mejor calidad.

Objetivo de DevOps

- **Automatizar** el ciclo de desarrollo y despliegue.
- **Reducir errores** en producción con pruebas y monitoreo.
- **Acelerar** la entrega de software con CI/CD.
- **Mejorar la colaboración** entre desarrolladores y operaciones.

¿Qué es CI/CD?

Práctica de DevOps que permite automatizar el desarrollo, pruebas y despliegue de software. Se divide en dos partes principales

CI (Integración Continua - Continuous Integration)

- Los desarrolladores suben cambios al código varias veces al día.
- Se ejecutan pruebas automáticas para detectar errores de inmediato.
- Asegura que el código nuevo funcione bien con el código existente.

CD (Entrega Continua - Continuous Delivery)

- El código está listo para producción, pero requiere aprobación manual para desplegarse.

CD (Implementación Continua - Continuous Deployment)

- El código se despliega automáticamente en producción sin intervención manual.

Entrega continua vs. Implementación continua

- La implementación continua es el mejor ejemplo de automatización de DevOps . Eso no significa que sea la única forma de realizar CI/CD, ni la forma “correcta”. Dado que la implementación continua depende de herramientas de prueba rigurosas y de una cultura de prueba madura, la mayoría de los equipos de software comienzan con la entrega continua e integran más pruebas automatizadas con el tiempo.

Análisis de datos – CI/CD

- **Implementación de pruebas:**

- - Pruebas unitarias de integración y de salida **(Hecho)**
- - Pruebas automáticas **(En progreso por compatibilidad con workflow de GitHub Actions)**

- **Integración continua:**

- - Crear workflow CI/CD en GitHub actions con .yaml **(Hecho)**
- - Ejecutar pruebas unitarias es cada push **(Hecho)**
- - Analizar código con SonarQube para detectar problemas **(Hecho)**
- - Construir imágenes Docker después de las pruebas **(Pendiente)**
- - Publicar artefactos en un registro de contenedores **(Pendiente)**

- **Seguridad y cumplimiento:**

- -Autenticación segura con cifrado Hash **(Hecho)**
- -Escaneo de vulnerabilidades con las alertas de GitHub **(Pendiente)**

Cambios en el repositorio

- **Commit 4b474e6:** Archivo inicial del proyecto
- **Commit c456b16:** Se creó la clase paquetes con atributos privados y la clase para crear los paquetes
- **Commit 23bf6e9:** Inicio de la creación de la gestión de usuarios
- **Commit 252a098:** Modificación en la creación de usuario
- **Commit 0bccc2a:** Creación de la base de datos y adición de datos de prueba al Excel
- **Commit 8f0ac9b:** Se añadió la base de datos que es un archivo xlsx y se hicieron pruebas de datos (gestión_p.py – paquetes.xlsx)
- **Commit 21dede0:** Se creó la variable *estado_pedido* en la base de datos y a la hora de crearse se vea como pendiente (gestión_p.py – paquetes.xlsx)

Cambios en el repositorio

- **Commit 5b98526:** Se continuó con la gestión de usuario. Se comenzó con el menú de cada tipo de cliente
- **Commit 2f63be7:** Se empezó a crear la gestión de envío creando la función *hacer_pedido*
- **Commit cc7812e:** Se incrementó la clase paquetes, ampliando la base de datos y se creó la función **actualizar_datos**
- **Commit cb251d4:** Agregando pipeline de GitHub Actions
- **Commit 98bceef:** Se hicieron cambios de varias funciones que estaban generando errores. Se cambió la función de **actualizar_datos**
- **Commit 700bd9d:** Se creó una variable para imprimir opción inválida



Universidad Tecnológica de Bolívar

Gracias

Copyright © 2023. Todos los derechos reservados Universidad Tecnológica de Bolívar.