

Evidencia De Aprendizaje 2. Optimizando La Productividad En El Mundo Del Software

Programación para Análisis de Datos (PREICA2501B020065)

Indira Johanna Hamdam Jarava

Giordan Jese Ricardo Parra

Docente

Andrés Felipe Callejas

Ingeniería de Software y Datos

Institución Universitaria Digital de Antioquia

2025

INTRODUCCIÓN

En el contexto del desarrollo moderno de software, la implementación de metodologías DevOps y el control de versiones con herramientas como Git y GitHub son fundamentales para garantizar la eficiencia, la colaboración y la entrega continua de productos. Esta evidencia de aprendizaje da continuidad al trabajo iniciado en la Unidad 1, ampliando sus alcances mediante el uso de flujos de trabajo DevOps, pruebas automatizadas y despliegue continuo.

El presente proyecto tiene como propósito documentar y demostrar la aplicación práctica del versionamiento de código, automatización con GitHub Actions y la virtualización del entorno de desarrollo, elementos esenciales para optimizar la entrega de soluciones tecnológicas. La virtualización fue un componente clave, ya que permitió configurar entornos estandarizados y reproducibles para asegurar que el software funcione de manera consistente en distintas máquinas y etapas del pipeline de desarrollo.

Objetivos

Objetivo general:

Implementar un flujo de trabajo DevOps que permita gestionar versiones, automatizar pruebas y realizar despliegues continuos de un proyecto de scraping utilizando Git, GitHub y GitHub Actions, en un entorno virtualizado.

Objetivos específicos:

- Configurar un repositorio en GitHub para el control de versiones del proyecto.
- Documentar y mantener actualizado el historial del desarrollo mediante Git.
- Establecer un entorno de desarrollo virtualizado para pruebas y ejecución del proyecto.
- Implementar un pipeline de CI/CD utilizando GitHub Actions que integre pruebas automatizadas y despliegue continuo.
- Documentar el flujo DevOps empleado, detallando las herramientas, configuraciones y procesos involucrados.

Descripción del Tema

En esta etapa del proyecto se continúa trabajando con el tema de extracción automatizada de datos desde sitios web, enfocándose esta vez en la organización del proyecto bajo prácticas profesionales. El objetivo es desarrollar un pipeline de DevOps que integre Git para el control de versiones, GitHub como repositorio central y GitHub Actions para la automatización de tareas como pruebas, despliegues y validación del código.

La infraestructura de virtualización se utiliza para simular un entorno de servidor, donde se desplegará la aplicación. Esto permite asegurar que el código que funciona en el entorno de desarrollo también lo haga correctamente en producción.

Metodología Empleada

Para el desarrollo de esta evidencia, se siguieron los siguientes pasos metodológicos:

1. Configuración del entorno de desarrollo:

Se utilizó un entorno virtualizado para garantizar que las dependencias y condiciones del entorno se mantuvieran constantes, independientemente del sistema operativo o configuración de la máquina del desarrollador.

2. Inicialización del proyecto con Git:

Se creó un repositorio Git local y posteriormente se vinculó a un repositorio remoto en GitHub. A partir de este punto, todo el código y la documentación se gestionó utilizando commits, ramas y pull requests.

3. Implementación del pipeline DevOps:

Se configuró un archivo .yml en el directorio .github/workflows del repositorio, donde se definieron los pasos del flujo de integración y entrega continua. Esto incluyó:

- Instalación de dependencias.
- Ejecución de pruebas automatizadas (si aplica).
- Validación del código.
- Despliegue automático a un entorno virtual o servicio web.

4. Documentación:

Cada etapa del desarrollo y configuración del pipeline fue documentada en el README del repositorio, incluyendo instrucciones para clonar, instalar dependencias, ejecutar el proyecto y entender el flujo CI/CD.

Resultados

Durante el desarrollo de esta evidencia, se logró implementar exitosamente un flujo de trabajo DevOps básico, empleando Git y GitHub como herramientas principales de control de versiones y colaboración. El repositorio fue correctamente configurado, incluyendo ramas de desarrollo, documentación en formato README, y un historial claro de cambios a través de commits.

Se incorporó una configuración funcional de GitHub Actions para automatizar procesos clave del ciclo de vida del proyecto. El pipeline de integración continua (CI) incluía verificación del código y validación del entorno, mientras que el pipeline de entrega continua (CD) permitía desplegar automáticamente el proyecto en un entorno virtualizado después de cada fusión a la rama principal.

Además, se utilizó virtualización como soporte de entorno de ejecución mediante contenedores o máquinas virtuales, lo que permitió asegurar que el comportamiento del software fuera consistente entre desarrollo y producción.

Se obtuvieron los siguientes logros concretos:

- Gestión de versiones organizada y trazable.
- Automatización de tareas mediante GitHub Actions.
- Documentación clara del proceso y del proyecto.
- Reproducibilidad del entorno de trabajo a través de la virtualización.

ENLACE

❖ https://github.com/Lashkmy/Analizando_EA1.git

CONCLUSIONES

La implementación del control de versiones con Git y GitHub, en conjunto con un pipeline automatizado mediante GitHub Actions, representó una mejora significativa en la profesionalización del desarrollo del proyecto.

Este enfoque permitió reducir errores humanos, estandarizar los procesos de integración y despliegue, y asegurar la continuidad del trabajo sin depender de configuraciones manuales. A su vez, el uso de entornos virtualizados demostró ser una práctica fundamental para garantizar que el software se ejecute correctamente en cualquier sistema, evitando problemas de compatibilidad.

Como conclusión general, esta evidencia reafirma la importancia de adoptar metodologías modernas de desarrollo, como DevOps, desde las etapas formativas, permitiéndonos adquirir competencias que serán fundamentales en proyectos reales, tanto personales como profesionales. Esta experiencia sirve como una base sólida para futuras implementaciones que incluyan pruebas más complejas, monitoreo y escalado de servicios.

BIBLIOGRAFÍA

- ✓ Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress. <https://git-scm.com/book/en/v2>
- ✓ GitHub Docs. (s.f.). *Understanding GitHub Actions*. Recuperado de <https://docs.github.com/en/actions>
- ✓ Python Software Foundation. (s.f.). *Python Documentation*. <https://docs.python.org/3/>
- ✓ W3Schools. (s.f.). *HTML, CSS and Python Tutorials*. <https://www.w3schools.com/>
- ✓ Crummy, L. (s.f.). *BeautifulSoup Documentation*. <https://www.crummy.com/software/BeautifulSoup/>