# **Evidencia de aprendizaje 3**

**Automatización y Despliegue con GitHub Actions**

**Edgar Alberto Cano Castaño**

**CC 70927752**

**PREICA2402B020101**

**Docente:**

## **Andrés Felipe Palacio Angel**

**Curso:**

## **Programación para Análisis de Datos**

**Programa Ingeniería de Software y Datos**

**Facultad Ingenierías y Ciencias Agropecuarias**

**Institución Universitaria Digital de Antioquia**

**2024**

**INTRODUCCIÓN**

**Este informe documenta el uso de GitHub Actions para automatizar procesos clave en el desarrollo de software. Como parte de las evidencias de aprendizaje de las Unidades 1 y 2, esta actividad integra el control de versiones con Git y GitHub, y desarrolla un pipeline de integración y entrega continua (CI/CD). Finalmente, se despliega una aplicación web simple en un entorno de alojamiento en la nube, demostrando cómo la metodología DevOps puede optimizar la entrega de servicios.**

### **Objetivos**

Objetivo general:  
Automatizar los procesos de desarrollo y despliegue de una aplicación web mediante GitHub Actions, asegurando una entrega eficiente y escalable.

Objetivos específicos:

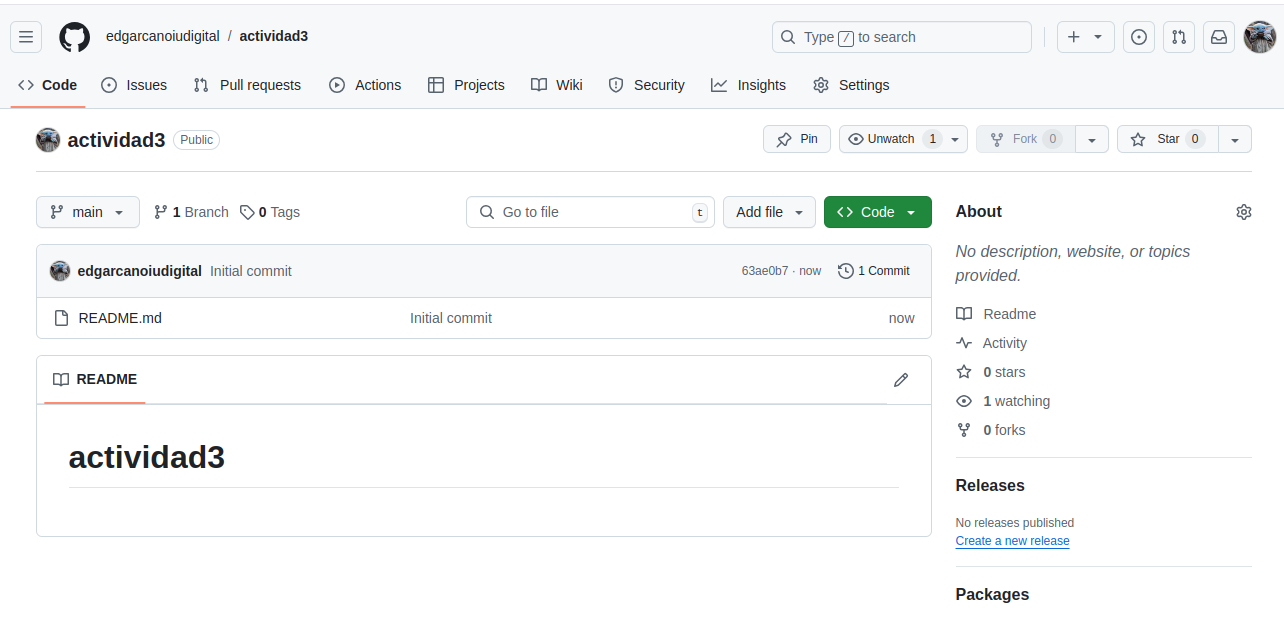
1. Diseñar una aplicación web simple para demostrar un flujo DevOps completo.
2. Configurar un pipeline de CI/CD en GitHub Actions que ejecute pruebas y despliegue automático.
3. Documentar el flujo de trabajo y los pasos técnicos implementados en un repositorio de GitHub.
4. Validar el despliegue en un entorno real, asegurando su correcto funcionamiento.

#### **Metodología Empleada**

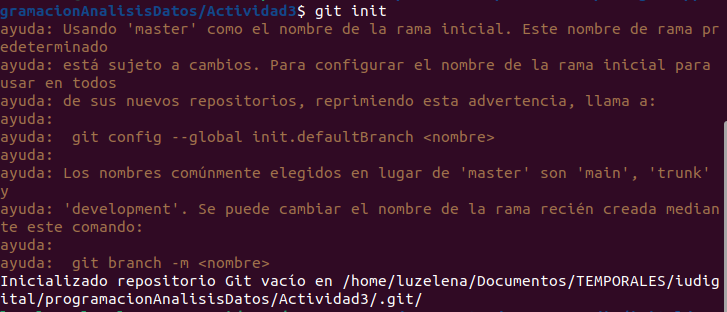
Fases del proceso:

1. Configuración del repositorio:
   * Creación del repositorio en GitHub.
   * Estructuración inicial del proyecto con archivos README, CONTRIBUTING y .gitignore.
2. Desarrollo del proyecto:
   * Utilización de Git para realizar commits frecuentes y documentados.
   * Implementación de las funcionalidades base del proyecto.
3. Automatización del flujo DevOps:
   * Configuración de GitHub Actions para CI/CD.
   * Definición de workflows para:
     + Ejecución de pruebas unitarias y de integración.
     + Generación de artefactos para despliegue.
     + Despliegue automático en [entorno elegido: AWS, Heroku, GitHub Pages, etc.].
4. Pruebas y validación:
   * Pruebas manuales y automatizadas para verificar la funcionalidad.
   * Resolución de errores detectados en el pipeline.
5. Documentación:
   * Creación de la documentación del proyecto, incluyendo instrucciones para ejecutar y mantener el flujo DevOps.

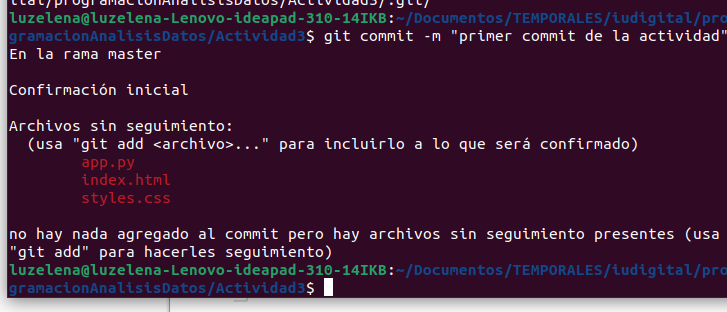
**Creación nuevo Repositorio:**

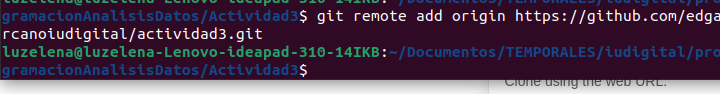


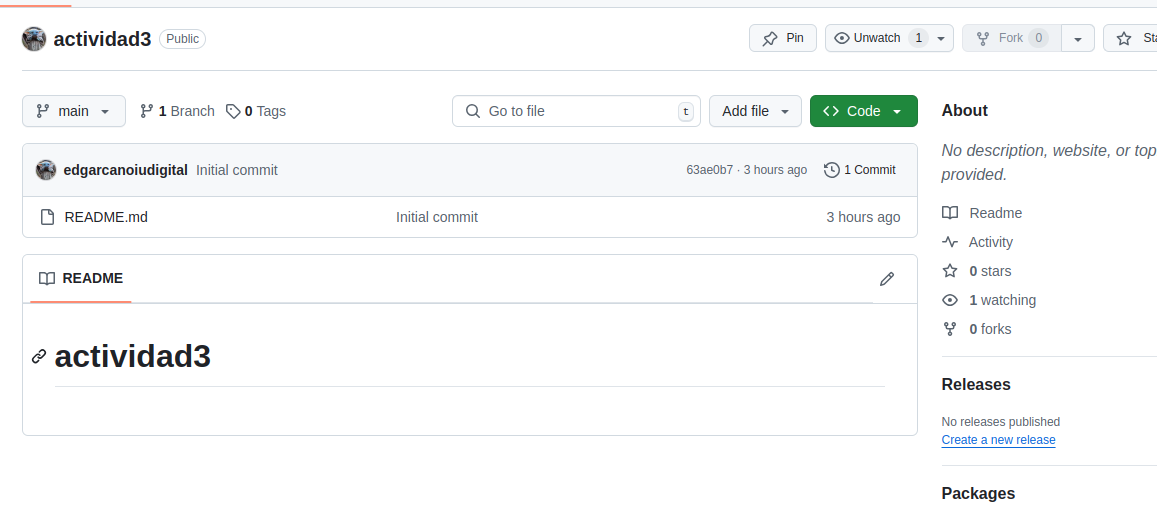
**Inicializar Repositorio**



**Primer Commit:**

**Configuramos Repositorio Remoto:**





Resultados obtenidos:

* Repositorio funcional en GitHub con un pipeline completo de CI/CD.
* Despliegue automatizado exitoso en [entorno seleccionado].
* Documentación detallada del flujo de trabajo DevOps y el proyecto desarrollado.

**URL Repositorio:**

[**https://github.com/edgarcanoiudigital/actividad3.git**](https://github.com/edgarcanoiudigital/actividad3.git)

#### **Bibliografía:**

* GitHub. (s.f.). GitHub Actions. Recuperado de: <https://github.com/features/actions>

**CONCLUSIONES:**

La implementación del flujo DevOps permitió automatizar tareas repetitivas y críticas, como las pruebas y el despliegue, mejorando la calidad y reduciendo los tiempos de entrega del software. Las herramientas utilizadas, especialmente Git y GitHub Actions, demostraron ser eficientes y prácticas para proyectos de desarrollo ágil.

**BIOGRAFÍA**

**Institución Universitaria Digital de Antioquia. (n.d.). Recuperado de** [**http://www.iudigital.edu.co**](http://www.iudigital.edu.co/)

**GitHub. (s.f.). GitHub Actions. Recuperado de: https://github.com/features/actions**