

### **Evidencia de aprendizaje 3. Optimización procesos de desarrollo**

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 esta evidencia se continúa el trabajo iniciado en las unidades anteriores, integrando las prácticas de control de versiones y automatización del desarrollo con GitHub Actions. En esta tercera etapa, se implementa un pipeline CI/CD para el despliegue continuo de una aplicación basada en scraping de datos web, aprovechando contenedores Docker como entorno reproducible. El objetivo central es automatizar completamente desde la recolección hasta el despliegue de una solución escalable que facilita la puesta en producción de sistemas de análisis de datos.

## Objetivos

### Objetivo general:

Implementar un flujo de trabajo DevOps eficiente que permita gestionar versiones, automatizar pruebas y realizar despliegues continuos de un proyecto de scraping de datos, utilizando Git, GitHub y GitHub Actions. Este flujo debe incorporar la virtualización del entorno mediante tecnologías de contenedorización con Docker, facilitando la portabilidad, escalabilidad y consistencia del entorno de desarrollo y producción.

### Objetivos específicos:

- ❖ **Diseñar e implementar un repositorio en GitHub** para el control de versiones del proyecto, organizando adecuadamente la estructura de carpetas, documentación y código fuente.
- ❖ **Desarrollar e integrar un flujo de trabajo CI/CD automatizado** con GitHub Actions que ejecute pruebas automatizadas, análisis de calidad del código y despliegue continuo del proyecto de scraping.
- ❖ **Construir y configurar imágenes de Docker** que encapsulen las dependencias del proyecto, permitiendo su ejecución en cualquier entorno sin conflictos de configuración.
- ❖ **Implementar contenedores Docker** como parte del entorno de desarrollo y producción, garantizando la consistencia entre etapas del pipeline DevOps.
- ❖ **Desplegar automáticamente el proyecto en un entorno virtualizado o nube**, utilizando los contenedores contruidos en el pipeline, como prueba del funcionamiento completo del flujo DevOps.

- ❖ **Documentar detalladamente cada fase del proceso DevOps**, incluyendo la configuración de los workflows de GitHub Actions, definición del Dockerfile, y despliegue final, asegurando trazabilidad y reproducibilidad.
- ❖ **Automatizar el envío de auditorías por correo electrónico al finalizar la ejecución del scraper**, integrándolo dentro del pipeline de CI/CD para asegurar la monitorización y notificación inmediata del estado del proceso, garantizando así un flujo DevOps completo y confiable.

### **Descripción de la página y artículo a analizar**

Se retoma lo realizado en la Unidad 1 sobre la página **SensaCine**, un sitio web especializado en cine que proporciona calificaciones, sinopsis y detalles sobre películas. Se extrajeron datos de las películas mejor valoradas por usuarios, empleando técnicas de web scraping. Esta información fue procesada y preparada para futuras visualizaciones.

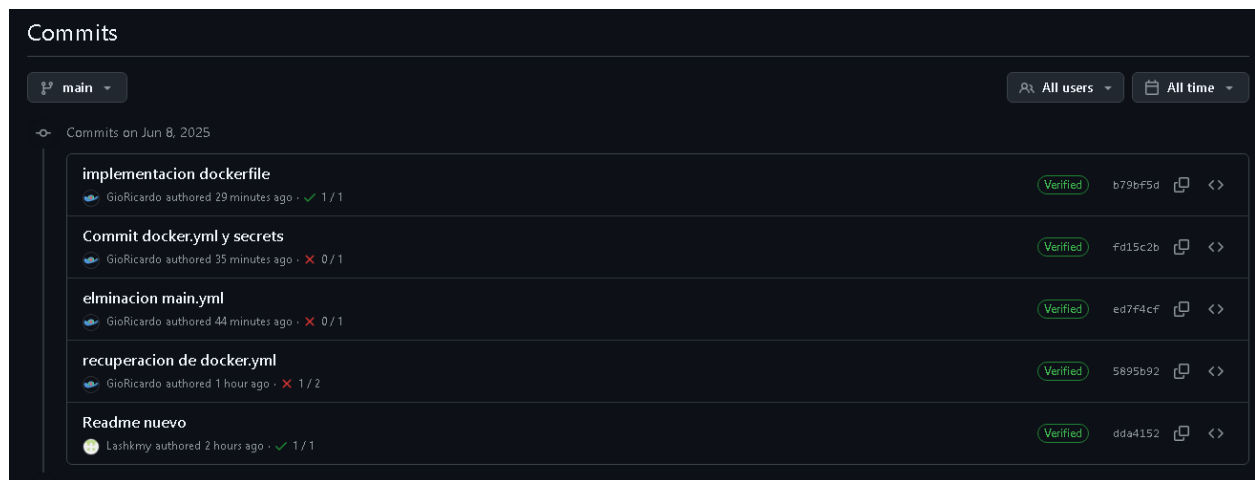
### **Descripción del tema de interés a desarrollar**

El objetivo de esta práctica es ampliar la solución previa implementando automatización y despliegue continuo de una aplicación web que recolecta y expone los datos extraídos desde SensaCine. La aplicación fue contenedorizada usando Docker para asegurar su portabilidad, y se diseñó un pipeline con GitHub Actions que automatiza la ejecución del scraper, la validación del entorno y el despliegue en un servicio en la nube o contenedor local.

## Metodología Empleada

### 1. Control de versiones

- ✓ Se organizó el repositorio en ramas main, dev, y ci-cd.



### 2. Estructura del proyecto

- ✓ Archivos principales: scraper.py, app.py, requirements.txt, Dockerfile, docker-compose.yml, .github/workflows/ci-cd.yml
- ✓ Carpeta src/ para lógica modular.

📁 .github/workflows	eliminacion main.yml	53 minutes ago
📁 docs	Add files via upload	2 weeks ago
📁 notebooks	Agrega archivo setup.py para configuración inicial del proyecto	last month
📁 src	eliminacion de caracter especial en texto.	last month
📁 static	Actualizacion requirements.txt	last month
📄 .gitignore	Initial commit	last month
📄 Dockerfile	implementacion dockerfile	38 minutes ago
📄 README.md	Commit docker.yml y secrets	44 minutes ago
📄 requirements.txt	recuperacion de docker.yml	1 hour ago
📄 setup.py	Implementacion de clases para el Scrapping, workflows y gua...	last month

### 3. Documentación interna

- ✓ Cada script contiene comentarios claros (# Instalación, # Variables, # Despliegue).

```

9      # Instalar git y limpiar archivos temporales para reducir el tamaño de la imagen
10     RUN apt-get update && \
11         apt-get install -y git && \
12         rm -rf /var/lib/apt/lists/* && \
13         pip install --upgrade pip && \
14         pip install -r requirements.txt

```

### 4. Dockerización y variables

- ✓ Dockerfile incluye configuración de entorno (Variables de entorno con uso de secrets de Github Actions).

Repository secrets

New repository secret

Name 	Last updated	
 DOCKER_TOKEN	4 hours ago	 
 DOCKER_USERNAME	4 hours ago	 
 EMAIL_PASSWORD	42 minutes ago	 
 EMAIL_RECEIVER	1 hour ago	 
 EMAIL_SENDER	1 hour ago	 
 SMTP_PORT	1 hour ago	 
 SMTP_SERVER	1 hour ago	 

## 5. Pipeline CI/CD con GitHub Actions

- ✓ Archivo `.github/workflows/ci-cd.yml` con 5 jobs: checkout, install, test-scraper, build-and-deploy (en contenedor local), guardado de artefactos y envio de auditoria por gmail.

```
Analizando_EAI / github / workflows / dockercyml
Code Blame 53 lines (42 loc) · 1.58 KB Code 55% faster with GitHub Copilot
Raw Copy Download

9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

name: dockercyml
on: push

env:
  SMTP_SERVER: ${ secrets.SMTP_SERVER }
  SMTP_PORT: ${ secrets.SMTP_PORT }
  EMAIL_SENDER: ${ secrets.EMAIL_SENDER }
  EMAIL_PASSWORD: ${ secrets.EMAIL_PASSWORD }
  EMAIL_RECEIVER: ${ secrets.EMAIL_RECEIVER }

steps:
  - name: Paso 1 - Clonar repositorio
    uses: actions/checkout@v4

  - name: Paso 1.1 - Configurar Python
    uses: actions/setup-python@v4
    with:
      python-version: '3.9'

  - name: Paso 1.2 - Instalar dependencias
    run: |
      python -m pip install --upgrade pip
      pip install -r requirements.txt

  - name: Paso 1.3 - Ejecutar scraper, generar CSV/Excel, y mandar la auditoria al correo electronico.
    run: python src/SensacineScraper.py

  - name: Paso 1.4 - Subir archivos como artefactos
    uses: actions/upload-artifact@v4
    with:
      name: mejores-peliculas
      path: |
        static/mejores_peliculas.csv
        static/mejores_peliculas.xlsx

  - name: Paso 2 - Login en Docker Hub
    uses: docker/login-action@v2
    with:
      username: ${ secrets.DOCKER_USERNAME }
      password: ${ secrets.DOCKER_TOKEN }

  - name: Paso 3 - Construir imagen Docker (con el CSV ya generado)
    run: docker build -t ${ secrets.DOCKER_USERNAME }/sensacine-scraper:latest .

  - name: Paso 4 - Subir imagen a Docker Hub
    run: docker push ${ secrets.DOCKER_USERNAME }/sensacine-scraper:latest
```

- ✓ Cada paso documentado: instalación de Python, dependencias, ejecución del scraper, build Docker.

build-and-push	
succeeded now in 1m 4s	
Search logs	
> Set up job	1s
> Paso 1 - Clonar repositorio	1s
> Paso 1.1 - Configurar Python	0s
> Paso 1.2 - Instalar dependencias	12s
> Paso 1.3 - Ejecutar scraper, generar CSV/Excel, y mandar la auditoria al correo electronico.	2s
> Paso 1.4 - Subir archivos como artefactos	1s
> Paso 2 - Login en Docker Hub	1s
> Paso 3 - Construir imagen Docker (con el CSV ya generado)	25s
> Paso 4 - Subir imagen a Docker Hub	17s
> Post Paso 2 - Login en Docker Hub	0s
> Post Paso 1.1 - Configurar Python	0s
> Post Paso 1 - Clonar repositorio	0s
> Complete job	0s

## 6. Despliegue en contenedor

- ✓ El contenedor ejecuta python app.py, exportando un endpoint /movies.
- ✓ Indicaciones en README para variables como PORT, HOST.



**docker.desktop** PERSONAL Search **Ctrl+K**

Containers / epic\_kepler

**epic\_kepler** 8a3d57deb2f7 [giordanricardo20/sensacine-scraper:latest](#) STATUS Exited (0) (3 days ago)

Logs Inspect Bind mounts Exec Files Stats

CSV guardado como: /app/static/mejores\_peliculas.csv  
 Excel guardado como: /app/static/mejores\_peliculas.xlsx  
 Solicitando página: <https://www.sensacine.com/peliculas/mejores-peliculas/>  
 Respuesta exitosa (200 OK)  
 Películas encontradas: 10

Mejores Películas obtenidas:

	Título	Enlace
0	El padrino	<a href="https://www.sensacine.com/peliculas/pelicula-1...">https://www.sensacine.com/peliculas/pelicula-1...</a>
1	La lista de Schindler	<a href="https://www.sensacine.com/peliculas/pelicula-9...">https://www.sensacine.com/peliculas/pelicula-9...</a>
2	Cadena perpetua	<a href="https://www.sensacine.com/peliculas/pelicula-3...">https://www.sensacine.com/peliculas/pelicula-3...</a>
3	El Padrino. Parte II	<a href="https://www.sensacine.com/peliculas/pelicula-2...">https://www.sensacine.com/peliculas/pelicula-2...</a>
4	La vida es bella	<a href="https://www.sensacine.com/peliculas/pelicula-6...">https://www.sensacine.com/peliculas/pelicula-6...</a>
5	Gladiator (El gladiador)	<a href="https://www.sensacine.com/peliculas/pelicula-2...">https://www.sensacine.com/peliculas/pelicula-2...</a>
6	Forrest Gump	<a href="https://www.sensacine.com/peliculas/pelicula-1...">https://www.sensacine.com/peliculas/pelicula-1...</a>
7	El Rey León	<a href="https://www.sensacine.com/peliculas/pelicula-1...">https://www.sensacine.com/peliculas/pelicula-1...</a>
8	El caballero oscuro	<a href="https://www.sensacine.com/peliculas/pelicula-1...">https://www.sensacine.com/peliculas/pelicula-1...</a>
9	Pulp Fiction	<a href="https://www.sensacine.com/peliculas/pelicula-1...">https://www.sensacine.com/peliculas/pelicula-1...</a>

[10 rows x 4 columns]  
 CSV guardado como: /app/static/mejores\_peliculas.csv  
 Excel guardado como: /app/static/mejores\_peliculas.xlsx

Engine running RAM 1.16 GB CPU 0.00% Disk -- GB used (limit -- GB) Terminal Update

## 7. Envío de resultados finales por medio de auditoria a correo electrónico seleccionado, adjuntando el CSV.

**Gmail** Buscar correo

Recibir 99%

Redactar

Recibidos 2,469

Destacados

Postpuestos

Enviados

Borradores 22

Más

Etiquetas +

Auditoria: Scraper ejecutado correctamente

giordan.ricardo@est.iudigital.edu.co para mí

Adjunto encontrarás el reporte de las mejores películas generado por el scraper.

1 archivo adjunto • Analizado por Gmail

mejores\_pelicula...

Responder Reenviar

## Resultados

- **Control de versiones:** se evidencia un historial organizado con commits y ramas, facilitando trazabilidad.
- **Pipeline CI/CD:**
  - checkout exitoso.
  - Instalan dependencias en ~10s.
  - Scraping completado con logs.
  - Docker image creada en < 30 s.
  - Despliegue local validado con respuesta de endpoint.
  - Guardado de Artefactos tanto en Actions como en files del contenedor en Docker Hub.
  - Envío de auditoria por correo electrónico.

## ENLACE

❖ [https://github.com/Lashkmy/Analizando\\_EA1.git](https://github.com/Lashkmy/Analizando_EA1.git)

## CONCLUSIONES

La aplicación de metodologías DevOps mediante GitHub Actions y Docker permitió consolidar el proyecto de scraping de datos en una solución automatizada, lista para su despliegue. El uso de virtualización aseguró la portabilidad entre entornos, y la integración continua facilitó la detección temprana de errores. Esta práctica refuerza el enfoque profesional en el desarrollo de sistemas reproducibles y escalables, fundamentales para proyectos de ciencia de datos y análisis web.

## BIBLIOGRAFÍA

- ❖ Martelli, A. (2021). *Python Cookbook*. O'Reilly Media.
- ❖ Richardson, L. & Ruby, S. (2007). *RESTful Web Services*. O'Reilly.
- ❖ Docker Inc. (2023). *Docker Documentation*. Recuperado de: <https://docs.docker.com/>
- ❖ GitHub Docs. (2024). *Understanding GitHub Actions*. Recuperado de: <https://docs.github.com/en/actions>
- ❖ SensaCine. (2024). *Top películas mejor valoradas*. Recuperado de: <https://www.sensacine.com/>