



Coordinación de
Educación Abierta y a Distancia
VICERRECTORADO ACADÉMICO



CULTURA DIGITAL Y SOCIEDAD

Actividad Autónoma 3

Unidad 2: Herramientas y Metodologías en Ciencia de Datos



FACULTAD DE
Ingeniería

Nombres: Jhoffe Moreano Goyes

Fecha: 7/11/2025

Carrera: Ciencia de Datos e Inteligencia Artificial

Periodo académico: 2025-2S

Semestre: Tercero A

Informe Resumido del Proyecto

1. Introducción

El presente proyecto pone en práctica el flujo de trabajo colaborativo utilizando Git y GitHub para la gestión eficiente de un proyecto orientado a Ciencia de Datos. La funcionalidad desarrollada se basa en el script **Preprocesamiento_Iris.py**, elaborado en Python, el cual ejecuta un pipeline completo de limpieza y preparación del dataset de Iris. Dicho proceso contempla el tratamiento de valores faltantes, la transformación y codificación de variables categóricas, así como la normalización de los atributos numéricos, con el fin de obtener un conjunto de datos listo para ser empleado en la construcción y entrenamiento de modelos de Machine Learning, listos para análisis posteriores.

1.1 Objetivo

El objetivo del proyecto fue aplicar un flujo básico de Git y GitHub junto con un proceso sencillo de preprocesamiento de datos utilizando el dataset Iris.

1.2 Dataset Utilizado

El dataset Iris contiene medidas de pétalos y sépalos de tres especies de flores utilizado comúnmente para tareas de clasificación.

1.3 Enlace al Repositorio GitHub:

<https://github.com/0603916719/Preprocesamiento-Ciencia-Datos>

2. Comandos Git Utilizados

A continuación, se presentan los comandos principales de Git que se emplearon durante el desarrollo del objetivo y de este proyecto:

- git clone: Permite clonar el repositorio desde GitHub a mi computadora local.
- git checkout -b feature-preprocesamiento: Crea y cambia a una nueva rama para trabajar de manera ordenada.
- git add . : Agrega los cambios realizados al área de preparación.
- git commit -m 'Mensaje': Guarda los cambios localmente con una descripción clara.
- git push -u origin feature-preprocesamiento: Envía la rama con los cambios al repositorio remoto.
- git checkout main: Permite volver a la rama principal.
- git merge feature-preprocesamiento: Fusiona los cambios de la rama de trabajo con la rama principal.
- git branch -d feature-preprocesamiento: Elimina la rama local después de la fusión.

3. Automatización con GitHub Actions

Se configuró un workflow de **GitHub Actions** para automatizar la verificación del proyecto: se creó el archivo .github/workflows/ci.yml que se ejecuta en cada **push** y **pull request** hacia main, donde el runner realiza actions/checkout para obtener el código, actions/setup-python para preparar Python 3.11, instala dependencias con pip install -r Requirements_Iris.txt y ejecuta el script python Pocesamiento_Iris.py; con esto se valida automáticamente el preprocesamiento del dataset Iris y, si todo finaliza sin errores, la PR queda con **check verde** (éxito), proporcionando trazabilidad y asegurando reproducibilidad antes de fusionar cambios.

Control de Versiones con Git

- Se configuró el usuario de Git.
- Se creó y clonó un repositorio en GitHub.
- Se trabajó en una rama para preprocesamiento.
- Se realizó pull request y fusión hacia la rama principal.

GitHub Actions

Se configuró un workflow automatizado para ejecutar el script de preprocesamiento al realizar cambios en el repositorio, demostrando integración continua básica.

Comandos ejecutados dentro del workflow:

- Configuración del ambiente.
- Instalación de dependencias.
- Ejecución de scripts de preprocesamiento o pruebas automáticas.

Luego se realizó la pull request desde GitHub, solicitando revisión y confirmando la fusión. Finalmente, se verificó la ejecución exitosa del workflow.

4. Evidencia y Capturas de Pantalla

A continuación se muestran las capturas de pantalla del proceso:

- Clonación del repositorio.

```
user@MacBook-Air-de-USER ~ % git config --global user.name "Jhoffre Moreano"
git config --global user.email "peke_77moreano@hotmail.com"
git config --global credential.helper osxkeychain

user@MacBook-Air-de-USER ~ % git config --global --list

pull.rebase=true
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=Jhoffre Moreano
user.email=peke_77moreano@hotmail.com
credential.helper=osxkeychain
user@MacBook-Air-de-USER ~ %
```

Figura 1. Configuración Global de Usuario en Git.

La imagen muestra la configuración global del usuario en Git, donde se establece el nombre, el correo electrónico y el método de almacenamiento de credenciales, y luego se verifica dicha configuración con git config --global --list.

- Creación y push de la rama feature-preprocesamiento.

```
Preprocesamiento-Ciencia-Datos — zsh — 80x24
~/Preprocesamiento-Ciencia-Datos/Preprocesamiento-Ciencia-Datos — zsh +
```

```
.
```

	description	index	objects
..	HEAD	info	packed-refs
config	hooks	logs	refs

```
user@MacBook-Air-de-USER Preprocesamiento-Ciencia-Datos % git clone https://github.com/0603916719/Preprocesamiento-Ciencia-Datos.git

Cloning into 'Preprocesamiento-Ciencia-Datos'...
remote: Enumerating objects: 35, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 35 (delta 10), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (35/35), 16.08 KiB | 2.30 MiB/s, done.
Resolving deltas: 100% (10/10), done.
user@MacBook-Air-de-USER Preprocesamiento-Ciencia-Datos % cd Preprocesamiento-Ciencia-Datos

user@MacBook-Air-de-USER Preprocesamiento-Ciencia-Datos % git config --global credential.helper osxkeychain

user@MacBook-Air-de-USER Preprocesamiento-Ciencia-Datos % git checkout -b feature-preprocesamiento

Switched to a new branch 'feature-preprocesamiento'
user@MacBook-Air-de-USER Preprocesamiento-Ciencia-Datos %
```

Figura 2. Clonación del repositorio desde GitHub

La imagen muestra la clonación del repositorio desde GitHub, el ingreso a la carpeta del proyecto, la configuración del método de autenticación y la creación de una nueva rama llamada feature-preprocesamiento.

- Estado exitoso del workflow en GitHub Actions.

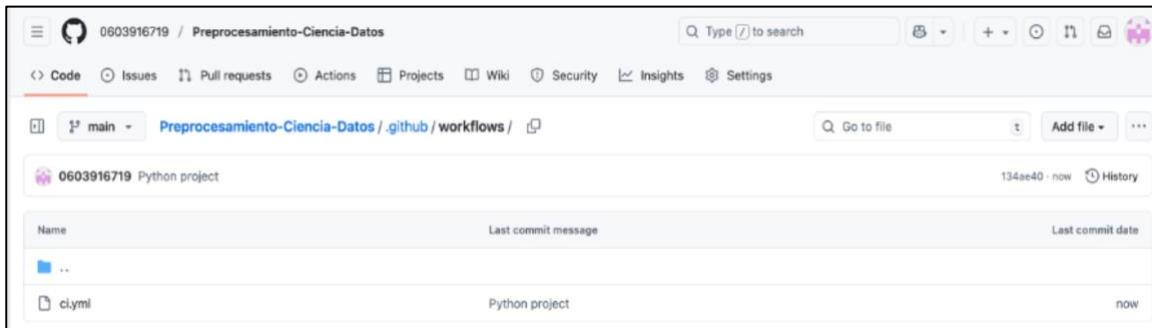
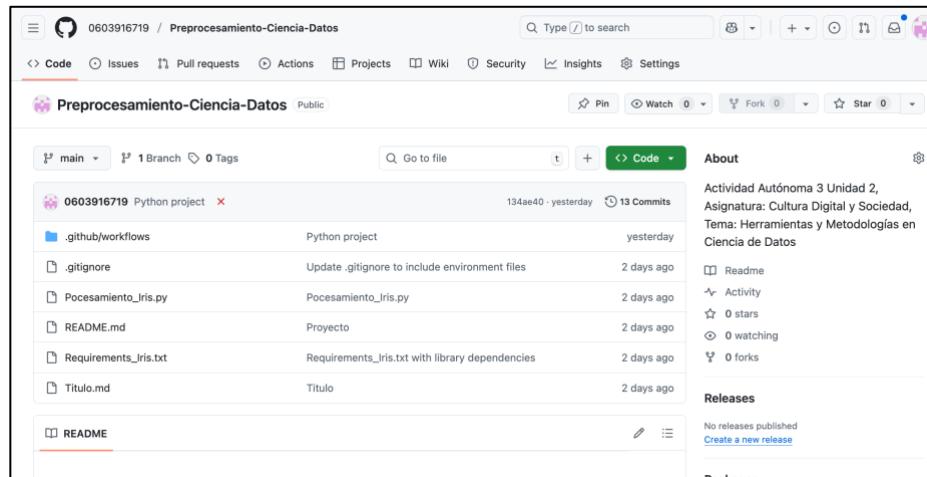


Figura 3. Creación del workflow de automatización en GitHub Actions

La imagen muestra la carpeta .github/workflows dentro del repositorio en GitHub, donde se encuentra el archivo ci.yml responsable de la automatización con GitHub Actions.

Configuración Commit

Se creó el repositorio en GitHub, se definió el archivo .gitignore para excluir archivos no necesarios y se efectuó el primer commit incluyendo el archivo inicial README.md.



The screenshot shows a GitHub repository page. On the left, the 'Files' tab is selected, displaying the project structure:

```

main
  ├── github
  │   ├── .gitignore
  │   └── Preprocesamiento_Iris.py
  ├── README.md
  ├── Requirements_Iris.txt
  └── Titulo.md

```

The main content area is titled 'Preprocesamiento de Datos del Iris' and contains sections like 'OBJETIVO' and 'Estructura de Carpetas'. The 'Estructura de Carpetas' section shows the following directory tree:

```

preprocesamiento-ciencia-datos/
├── .gitignore          # Archivos y carpetas a ignorar por Git
├── README.md           # Documentación principal del proyecto
├── data/
│   ├── raw/             # Datos originales
│   └── processed/       # Datos limpios listos para modelar
└── notebooks/
    └── exploracion_iris.ipynb # Análisis exploratorio y visualización
├── src/
│   └── preprocesamiento.py # Limpieza y transformación
└── reports/
    └── informe.md         # Informe final

```

Clonar el repositorio para usar el IDE Visual Studio Code

```

user — git-remote-https • git clone https://github.com/User/Preprocesamiento-Ciencia-Datos.git
...te-https • git clone https://github.com/User/Preprocesamiento-Ciencia-Datos.git
Last login: Fri Nov  7 10:06:44 on ttys000
user@MacBook-Air-de-USER ~ % cd ~
mv Preprocesamiento-Ciencia-Datos Preprocesamiento-Ciencia-Datos.backup
git clone https://github.com/User/Preprocesamiento-Ciencia-Datos.git

Cloning into 'Preprocesamiento-Ciencia-Datos'...
Username for 'https://github.com': "0603916719"
Password for 'https://0603916719@github.com': 

```

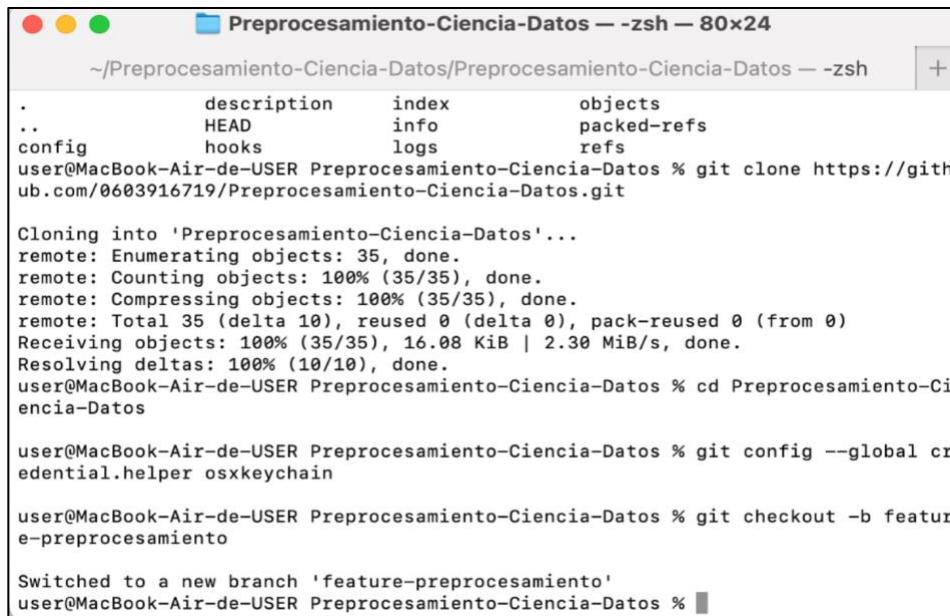
Validamos el usuario y correo de Git Hub

```

user — -zsh — 80x24
~ -- -zsh
Last login: Fri Nov  7 10:07:46 on ttys001
user@MacBook-Air-de-USER ~ % git config --global user.name
Jhoffre Moreano
user@MacBook-Air-de-USER ~ % git config --global user.email
peke_77moreano@hotmail.com
user@MacBook-Air-de-USER ~ % git config --global --list | grep user
user.name=Jhoffre Moreano
user.email=peke_77moreano@hotmail.com
user@MacBook-Air-de-USER ~ %

```

Desarrollo en Ramas



```
Preprocesamiento-Ciencia-Datos -- zsh -- 80x24
~/Preprocesamiento-Ciencia-Datos/Preprocesamiento-Ciencia-Datos -- zsh
.
..
config      description      index      objects
HEAD        hooks          info       packed-refs
hooks        logs           refs
user@MacBook-Air-de-USER Preprocesamiento-Ciencia-Datos % git clone https://github.com/0603916719/Preprocesamiento-Ciencia-Datos.git

Cloning into 'Preprocesamiento-Ciencia-Datos'...
remote: Enumerating objects: 35, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 35 (delta 10), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (35/35), 16.08 KiB | 2.30 MiB/s, done.
Resolving deltas: 100% (10/10), done.
user@MacBook-Air-de-USER Preprocesamiento-Ciencia-Datos % cd Preprocesamiento-Ciencia-Datos

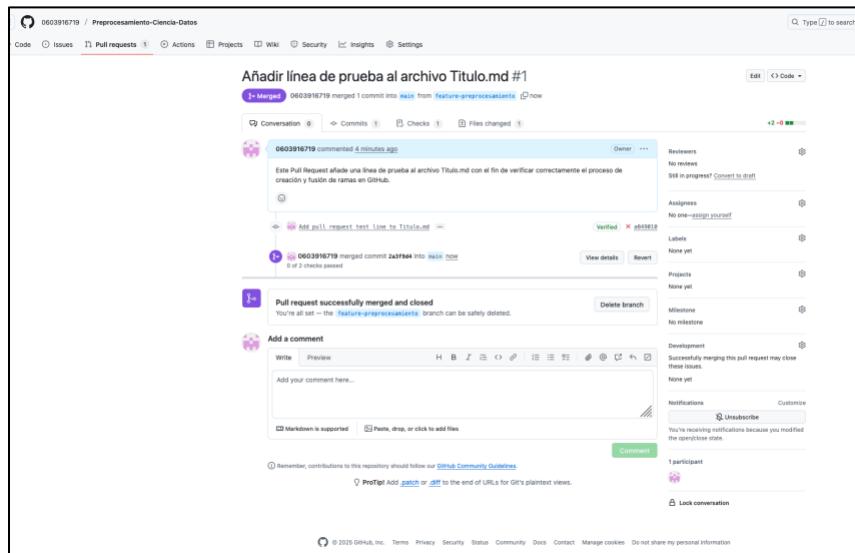
user@MacBook-Air-de-USER Preprocesamiento-Ciencia-Datos % git config --global credential.helper osxkeychain

user@MacBook-Air-de-USER Preprocesamiento-Ciencia-Datos % git checkout -b feature-preprocesamiento

Switched to a new branch 'feature-preprocesamiento'
user@MacBook-Air-de-USER Preprocesamiento-Ciencia-Datos %
```

Para incorporar el script de preprocesamiento, se generó una rama independiente denominada feature-preprocesamiento, con el fin de trabajar de forma aislada y evitar modificaciones directas en la rama principal main durante el proceso de desarrollo.

Pull Request y Fusión



El Pull Request titulado “**Añadir línea de prueba al archivo Titulo.md**” ha sido **fusiónado correctamente** desde la rama feature-preprocesamiento hacia main. La fusión se realizó exitosamente.

Conclusión

El proyecto permitió afianzar el uso de Git, GitHub y preprocesamiento básico de datos en un entorno controlado, logrando un flujo de trabajo claro y reproducible.

Bibliografía

- Git. (2025). *Git SCM: Documentation*. <https://git-scm.com/doc>
- Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress. <https://git-scm.com/book/en/v2>
- GitHub. (2025). *GitHub Docs: Understanding workflows*. <https://docs.github.com/actions>
- GitHub. (2025). *About pull requests*. <https://docs.github.com/pull-requests>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python*. *Journal of Machine Learning Research*, 12, 2825–2830. <https://scikit-learn.org/>
- Fisher, R. A. (1936). *The use of multiple measurements in taxonomic problems*. *Annals of Eugenics*, 7(2), 179–188. (Dataset Iris original)
- Seaborn. (2025). *Seaborn: Statistical Data Visualization*. <https://seaborn.pydata.org/>
- McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. *Proceedings of the 9th Python in Science Conference*, 51–56. (Biblioteca Pandas) <https://pandas.pydata.org/>
- Python Software Foundation. (2025). *Python Language Reference, version 3.11*. <https://www.python.org/>