Como **crear una rama para trabajor yo solo llamada develop/javier y sicronizar mi remoto y traerme a local lo de la rama main desdde la consola**, detalladamente, por favor?

Te explico paso a paso cómo crear y gestionar tu rama personal desde la consola:

1. Primero, asegúrate de estar en la rama main y tener todo actualizado:

bash

Copy

*# Ir a la rama main*

git checkout main

*# Actualizar el repositorio remoto*

git fetch origin

*# Traer cambios del remoto a tu main local*

git pull origin main

1. Crear tu rama personal desde main:

bash

Copy

*# Crear y cambiar a tu nueva rama*

git checkout -b develop/javier

El flag -b crea la rama y hace checkout a ella en un solo comando.

1. Subir tu nueva rama al repositorio remoto:

bash

Copy

*# Publicar tu rama en GitHub*

git push -u origin develop/javier

El flag -u establece el tracking, conectando tu rama local con la remota.

1. Verificar el estado:

bash

Copy

*# Ver en qué rama estás*

git branch

*# Ver todas las ramas (locales y remotas)*

git branch -a

1. Para mantener tu rama actualizada con main:

bash

Copy

*# Estando en tu rama develop/javier*

git fetch origin main

*# Traer los cambios de main*

git merge origin/main

🔑 Comandos adicionales útiles:

bash

Copy

*# Ver el estado de tus archivos*

git status

*# Ver las diferencias*

git diff

*# Ver el historial de commits*

git log --oneline

❗ Puntos importantes:

* Siempre haz pull de main antes de crear tu rama
* Resuelve conflictos en tu rama, no en main
* Mantén tu rama actualizada regularmente con main
* Haz commits frecuentes y con mensajes descriptivos

¿Necesitas que te explique algún paso con más detalle?

**WEBEX 12/02/2025**

**CICD**

crear los pipelines que hemos visto en este modulo y en 5, solo llegar hasta el presente para el futuro crearemos nuevas tareas:

1. Probar pipelines modulo 3 en c8, realizados los test.
2. Probar pipelines modulo 4 en c2, realizando los test.
3. Probar pipelines modulo 4 en c3, realizando los test.

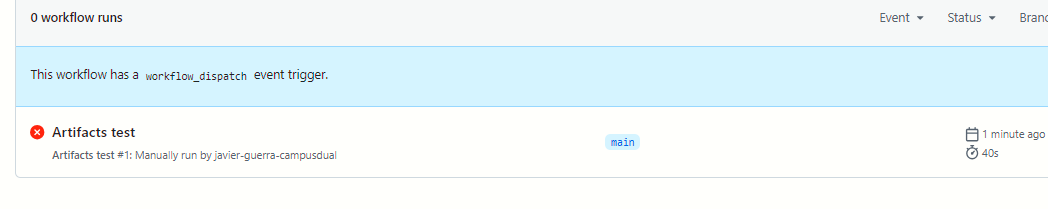
11/02/2025:

* Test pipelines modulo 3 en c8



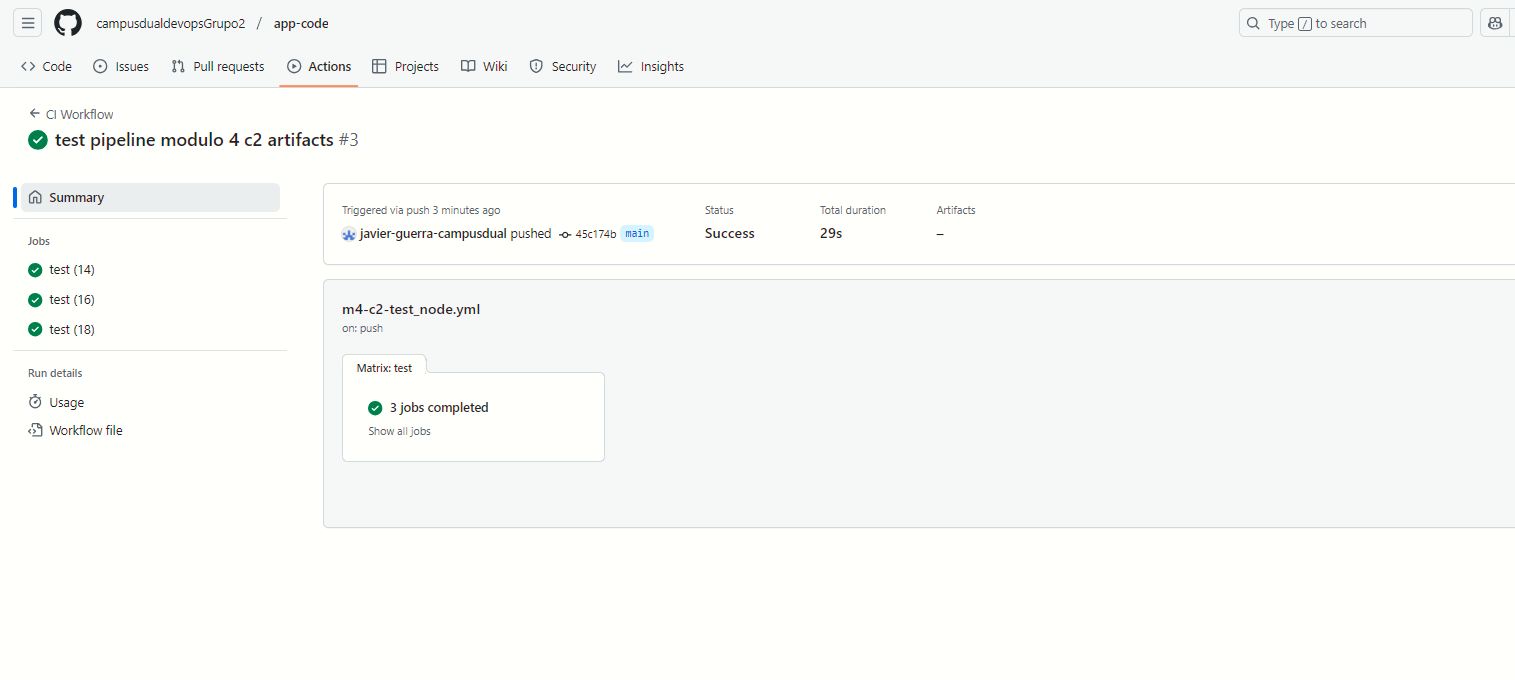
12/02/2025:

* Test pipelines modulo 4 en c2 test\_node





cat output-${{ matrix.node-version }}.json



crear los pipelines que hemos visto en este modulo y en 5, solo llegar hasta el presente para el futuro crearemos nuevas tareas:

1. Probar pipelines modulo 3 en c8, realizados los test.

2. Probar pipelines modulo 4 en c2, realizando los test.

3. Probar pipelines modulo 4 en c3, realizando los test.

11/02/2025:

- Test pipelines modulo 3 en c8

![Image](https://github.com/user-attachments/assets/1c0df670-e494-42eb-9bcd-4714ae2577d1)

12/02/2025:

- Test pipeline modulo 4 en c2, fallos en error:

![Image](https://github.com/user-attachments/assets/ef7e6382-4a21-441a-8c21-8e335487908a)

13/02/2025:

Debug:

- Verificación de fechas de expiración de los artefactos.

- Verificación de las ramas.

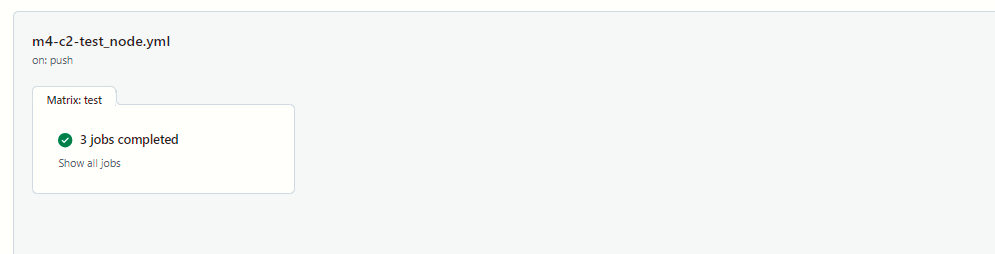
- Verificación de nombres.

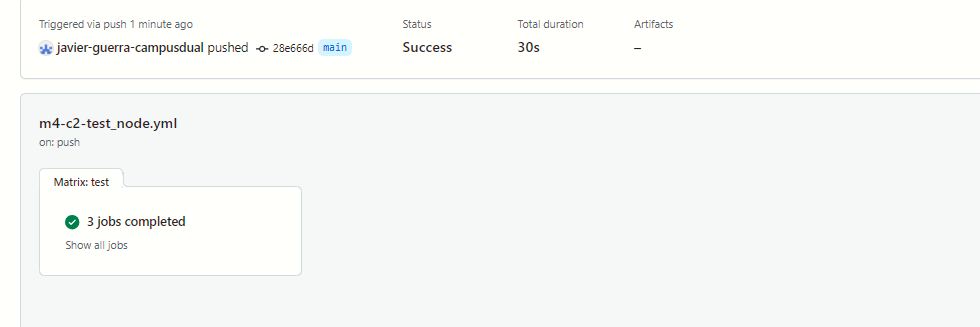
- Verificación de la ruta de los artefactos.

- Verificación de la creación de los artefactos.

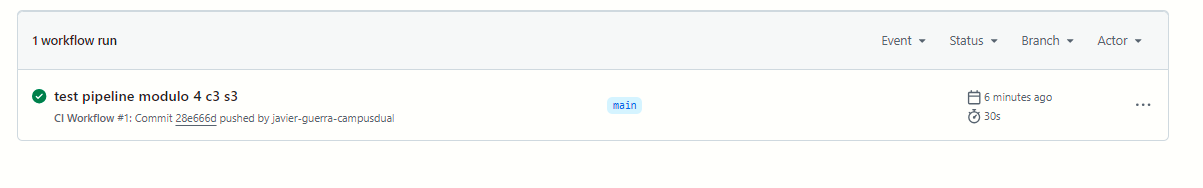
- Verificacíón de ejecucion de tests.

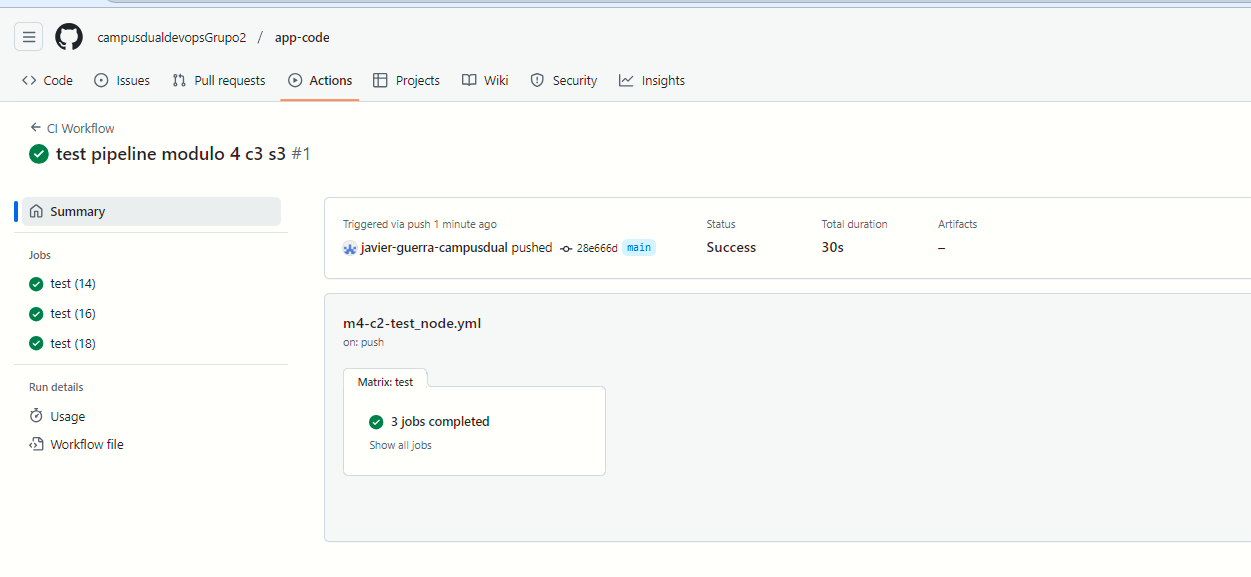
![Image](https://github.com/user-attachments/assets/ed422706-b190-44b3-83ac-1009500c08f5)





14/02/2025:





yaml

name: Artifacts test

on:

workflow\_dispatch:

jobs:

test:

runs-on: ubuntu-latest

strategy:

matrix:

node-version: [ 14, 16, 18 ]

steps:

- name: Checkout repository

uses: actions/checkout@v2

- name: Set up Node.js ${{ matrix.node-version }}

uses: actions/setup-node@v2

with:

node-version: ${{ matrix.node-version }}

- name: Install dependencies

working-directory: ./g2-cdt-m4-c1-node\_app

run: npm install

- name: Ejecutar tests

working-directory: ./g2-cdt-m4-c1-node\_app

run: npm test -- --json --outputFile=\_\_test\_\_/output-${{ matrix.node-version }}.json

- name: Subir resultados de tests como artefactos

uses: actions/upload-artifact@v4

with:

name: test-results-${{ matrix.node-version }}

path: ./g2-cdt-m4-c1-node\_app/\_\_test\_\_/output-${{ matrix.node-version }}.json # Corrige la ruta del archivo JSON

print-artifacts:

runs-on: ubuntu-latest

strategy:

matrix:

node-version: [ 14, 16, 18 ]

needs: test

steps:

- name: Descargar artefactos

uses: actions/download-artifact@v4

with:

name: test-results-${{ matrix.node-version }}

- name: Mostrar contenido del artefacto (output-${{ matrix.node-version