

GITHUB- **ACTIONS**

Índice de **C O N T E N I D O S**

01. Introducción

02. Descripción general de la
GitHub Action

03. Ubicación del workflow

04. Explicación paso a paso del workflow

05. Explicación detallada de los pasos

06. Ejecución de la action

1. INTRODUCCIÓN

¿Qué es GitHub Actions?

Es una plataforma de automatización integrada directamente en GitHub. Permite crear flujos de trabajo (workflows) personalizados que reaccionan a eventos en tu repositorio.

Básicamente, te ayuda a automatizar tareas repetitivas como compilar código, ejecutar pruebas o desplegar aplicaciones sin salir de GitHub.



2. DESCRIPCIÓN GENERAL DE LA GITHUB ACTION

¿Qué problema resuelve?

Implementa un flujo de Integración Continua (CI) que valida automáticamente cada cambio subido al repositorio. Asegura que el proyecto se puede instalar y que supera las pruebas antes de llegar a producción, evitando integrar errores ("bugs").

¿En qué tipo de proyectos se puede usar?

En proyectos basados en Node.js y JavaScript, tales como:

- Servidores Backend (Express, Fastify).
- Aplicaciones Frontend (React, Vue).
- Librerías NPM.

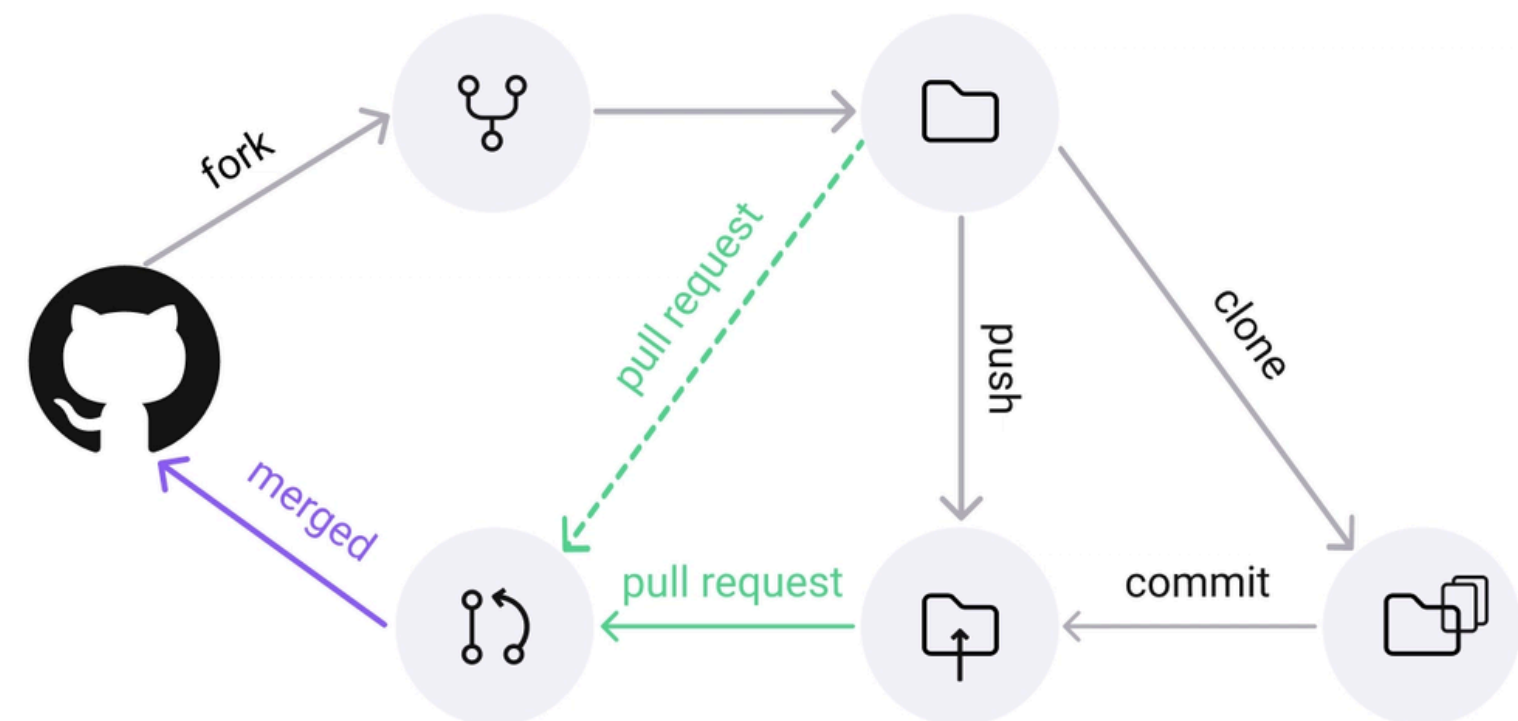
¿Por qué se ha elegido esta action?

Porque la automatización de pruebas es fundamental en la cultura DevOps. Elimina los errores humanos y soluciona el clásico problema de "en mi máquina funciona".

3. UBICACIÓN DEL WORKFLOW

- Ruta del directorio: .github/workflows/
- Archivo YAML: node-js.yml

GitHub Workflow



4. EXPLICACIÓN PASO A PASO DEL WORKFLOW

A continuación se analizan los componentes clave del archivo de configuración:

- name: Define el nombre visible del workflow en la pestaña Actions: "Node.js CI".
- on (Eventos):
 - push: Se dispara al subir cambios a la rama main.
 - pull_request: Se dispara al abrir una solicitud hacia la rama main.
- jobs: Contiene los trabajos que se ejecutan. En este caso hay un job llamado build-and-test.
- runs-on: Indica el sistema operativo del runner proporcionado por GitHub: ubuntu-latest.
- steps: Lista ordenada de pasos que se ejecutan dentro del job de forma secuencial.

Diferencia entre uses y run:

- uses: Invoca acciones predefinidas (como actions/checkout@v4).
- run: Ejecuta comandos de terminal estándar (como npm install).



5. EXPLICACIÓN DETALLADA DE LOS PASOS

A continuación se detalla la función de cada paso configurado en el workflow:

A. Checkout del código

Acción: `actions/checkout@v4` Descarga el código del repositorio y lo coloca en el entorno de trabajo del runner para poder trabajar con él.

B. Configurar Node.js

Acción: `actions/setup-node@v4` Instala Node.js (versión 20) y configura las variables necesarias para usar los comandos node y npm en el sistema.

C. Instalar dependencias

Comando: `npm install` Instala todas las dependencias definidas en el archivo `package.json`. Se usa `npm install` para dar flexibilidad en la práctica.

D. Ejecutar Tests

Comando: `npm test` Ejecuta las pruebas definidas en el proyecto. Éxito: Si tiene éxito, finaliza correctamente. Fallo: Si falla, marca el workflow como error y detiene el proceso.

6. EJECUCIÓN DE LA ACTION

¿Cuándo se ejecuta?

El flujo se activa automáticamente en dos situaciones:

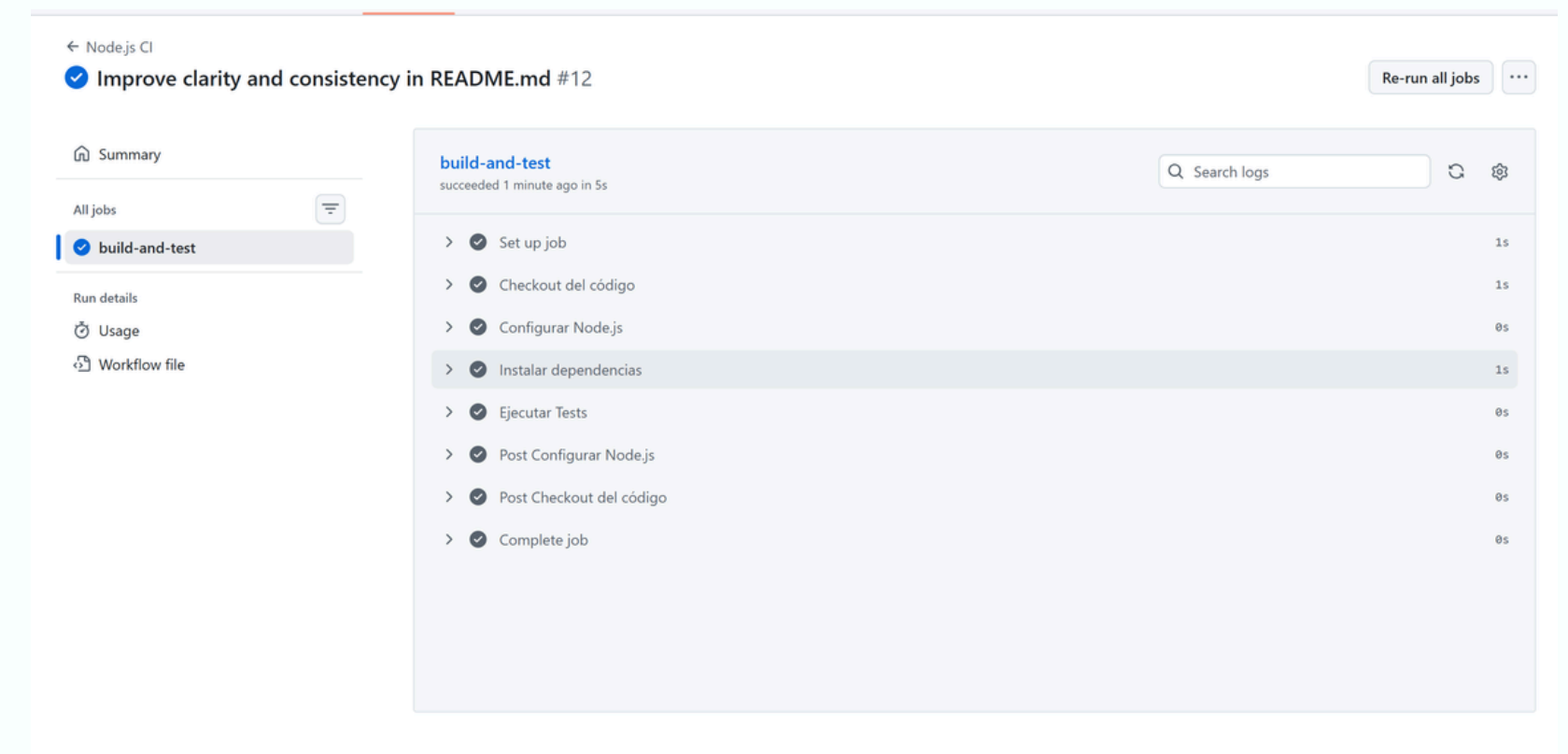
1. Cuando se hace push a la rama main.
2. Cuando se crea un pull request hacia main.

Evidencias de funcionamiento

Ejecución exitosa: Se verifica en los logs que el script de prueba (test.js) realiza la operación matemática correctamente.

```
> proyecto@1.0.0 test> node test.js
```

ÉXITO: 2 + 2 es 4



Enero 06

MUCHAS
GRACIAS

Parbjyot Singh