

# GITHUB ACTIONS

## Conceptos

### Workflow

## Workflow

- Proceso automatizado configurable que ejecutará uno o más jobs.
- Se define mediante un archivo YAML en el directorio `.github/workflows` de tu repositorio.

### Event

## Event

- Actividad específica en un repositorio, la cual activa una ejecución de un *workflow*.
- Puede originarse desde GitHub o de forma externa mediante un API REST.

### Job

## Job

- Conjunto de tareas (***steps***) en un *workflow* que se ejecutan en el mismo runner.
- Los *steps* se ejecutarán en orden y serán dependientes uno del otro.
- Como se ejecutan en el mismo *runner*, puedes compartir datos de un *step* a otro.

## Runner

### Runner

- Servidor que ejecuta los *workflows*.
- Puede ejecutar un *job* a la vez.
- GitHub provee runners de Ubuntu, Windows y MacOS.
- Puedes usar y configurar tus propios servidores como runners.

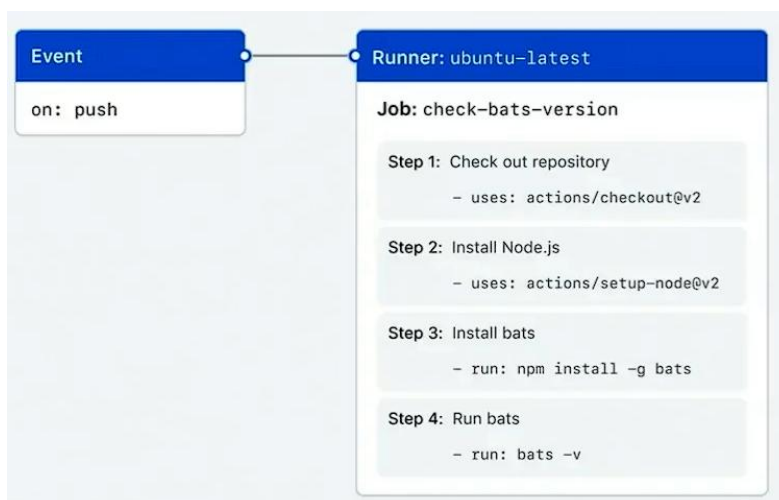


## Step

### Step

Puede ser:

- Un script de shell.
- Un *action* que se ejecutará.



## Action

# Action

- Aplicación personalizada que realiza una tarea compleja repetitiva.
- Ayuda a reducir código repetido.
- Se pueden usar Actions propias o utilizar Actions publicadas en el *GitHub Marketplace*.

## GitHub Marketplace

Types

Search for apps and actions

Sort: Best Match

Apps

Actions

Categories

API management

Chat

Code quality

Code review

Continuous integration

Dependency management

Deployment

IDEs

Learning

Localization

Mobile

Monitoring

Project management

Publishing

Recently added

Security

Support

Testing











Utilities

### Actions

An entirely new way to automate your development workflow.

17472 results filtered by Actions

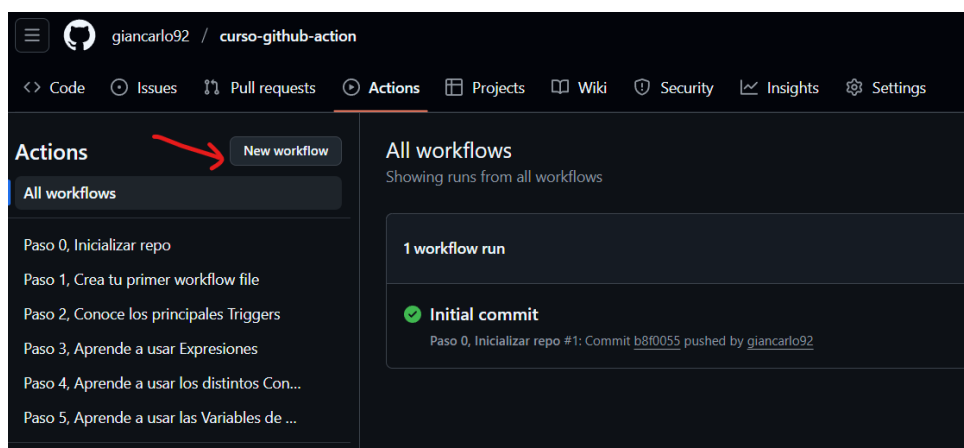
#### Actions

 <b>First interaction</b> By actions Greet new contributors when they create their first issue or open their first pull request ☆ 575 stars	 <b>Close Stale Issues</b> By actions Close issues and pull requests with no recent activity ☆ 910 stars
 <b>Setup .NET Core SDK</b> By actions Used to build and publish .NET source. Set up a specific version of the .NET and authentication to private NuGet repository ☆ 714 stars	 <b>Setup Java JDK</b> By actions Set up a specific version of the Java JDK and add the command-line tools to the PATH ☆ 1k stars
 <b>Download a Build Artifact</b> By actions Download a build artifact that was previously uploaded in the workflow by the upload-artifact action ☆ 891 stars	 <b>Upload a Build Artifact</b> By actions Upload a build artifact that can be used by subsequent workflow steps ☆ 2.2k stars
 <b>Setup Go environment</b> By actions Setup a Go environment and add it to the PATH ☆ 1k stars	 <b>Setup Node.js environment</b> By actions Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH ☆ 2.8k stars
 <b>Cache</b> By actions Cache artifacts like dependencies and build outputs to improve workflow execution time ☆ 3.5k stars	 <b>ServiceNow DevOps Register Artifact</b> By ServiceNow Register Artifact Details in ServiceNow

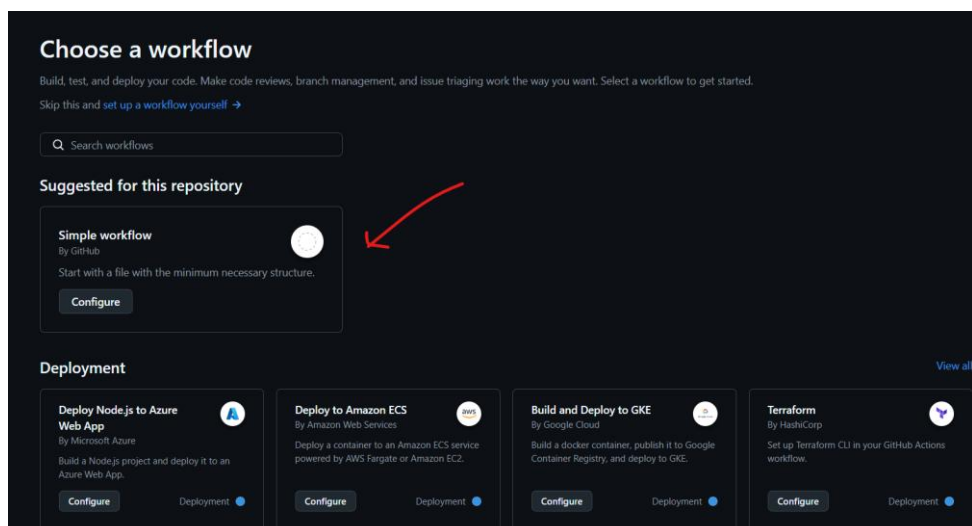
## Ejemplo de un YML con todos los pasos

```
name: hola-mundo
on: [push]
jobs:
  hola-mundo:
    runs-on: ubuntu-latest
    steps:
      - name: Public IP
        id: ip
        uses: haythem/public-ip@v1.3
      - name: Hola mundo
        run: echo ¡Hola Mundo desde ${steps.ip.outputs.ipv4}!
```

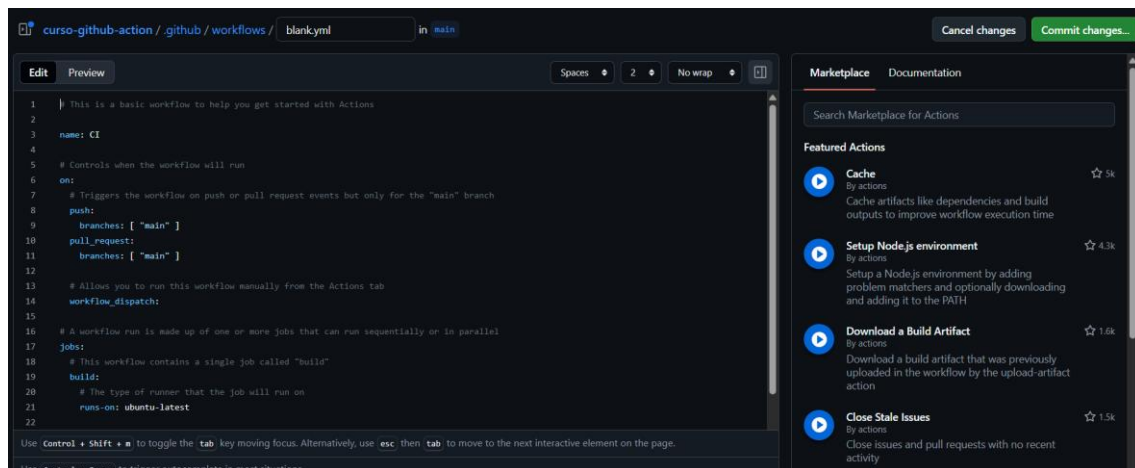
## Crear un nuevo workflow



## Plantilla básica

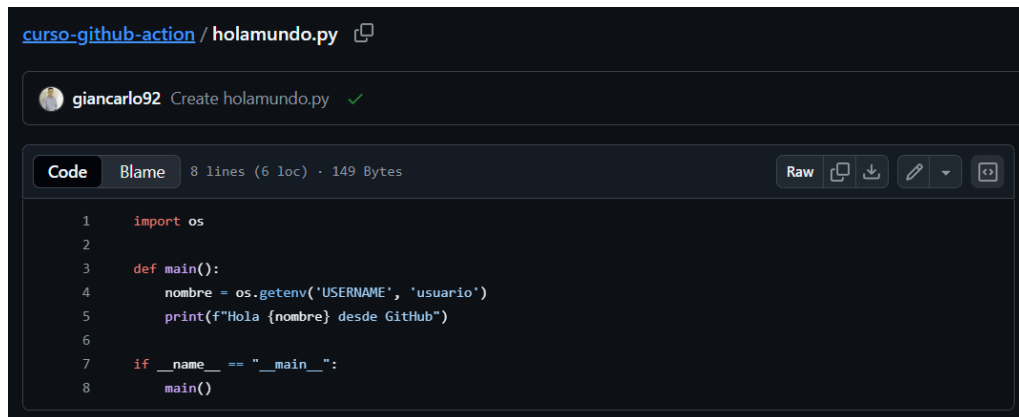


Esqueleto y acciones hechas por la comunidad



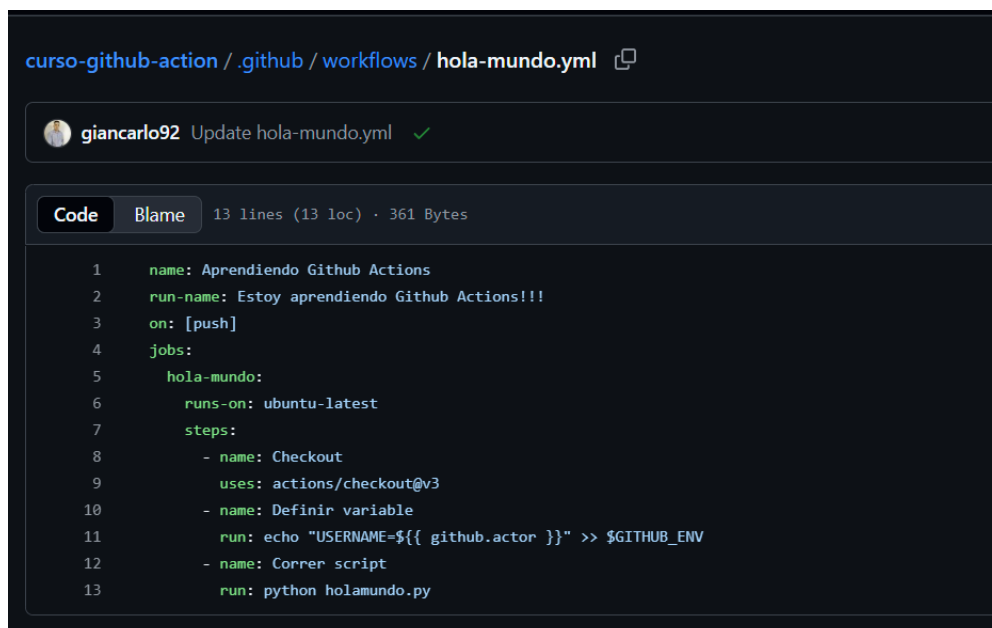
## Hola mundo en Python

Archivo py



Archivo yml (verificar la ruta)

Todos estos Jobs corren en el mismo espacio (servidor)



## Triggers

# Triggers

- push
- pull\_request
- issues

- issue\_comment
- workflow\_dispatch
- schedule

### Push

Push se entiende cuando mandamos al repositorio remoto las actualizaciones del repositorio local.

Este script acaparará el trigger push mientras suceda en los branches main o releases/\*\* (cualquier rama con nombre release o que nazca a partir de este nombre) y solo se tendrá en cuenta si se han modificado archivos de formato .js.

```
on: #Acaparará los triggers de este workflow
  push: #El tipo de trigger: push
    branches: #Las ramas en las que se activará el workflow
      - 'main' #Se activará si se hace push en la rama main
      - 'releases/**' #O en la rama releases y derivados
    paths: #Serán las rutas que se tendrán en cuenta para activar el workflow
      - '**.js' #Todos los archivos .js en el repositorio
```

### Pull Request

El Pull Request será la petición de mergear una rama aislada con otra rama (generalmente principal).

Este script se activará por un PR, específicamente en el momento de abrirlo y etiquetarlo (opened y labeled), seguirá el resto de reglas que el push

```
on: #Acaparará los triggers de este workflow
  pull_request: #El tipo de trigger: pr
    types: #Los estados que activarán el trigger
      - [opened, labeled] #Tomará los estados del PR
    branches: #Las ramas en las que se activará el workflow
      - 'releases/**' #En la rama releases y derivados
    paths: #Serán las rutas que se tendrán en cuenta para activar el workflow
      - '**.js' #Todos los archivos .js en el repositorio
```

## Issues

Los issues son foros donde se anuncian funcionalidades del código por parte de la comunidad o los propios developers.

Tiene la misma lógica de los PR, sin embargo, los types cambiarán.

```
on: #Acaparará los triggers de este workflow
  issues: #El tipo de trigger: issues
    types: #Los estados que activarán el trigger
      - [opened, edited, closed]
```

## Issue Comment

Cuando se generen comentarios en un Issue o un PR se activará este trigger.

Tendremos 2 casos, el primero donde se ejecutará un issue comment, y el segundo donde se ejecutará en un PR comment.

En este caso cada que se cree o elimine un comentari

```
on:
  issue_comment:
    types: [created, deleted]
```

En este segundo ejemplo también trabajaremos por issue\_comment, sin embargo, añadiremos un job que verificará que estamos sobre un pull request con el condicional dado, si es verdad, podrá continuar.

```
on: issue_comment
jobs:
  pr_commented:
    name : PR comment
    if : ${{ github.event.issue.pull_request }}
```

## Workflow dispatch

Los Workflow dispatch son Workflows de activación manual, estos poseen la característica de poder setear inputs, los configuraremos dentro del archivo YAML.

Este Workflow contendrá 3 variables a ingresar las cuales serán alerta, tags y enviroment, serán contenidas por la etiqueta inputs y en cada una especificaremos sus características. La primera tendrá una descripción, será obligatoria, tendrá medio como valor por defecto y será de tipo choice (las opciones serán alto, medio o bajo). Los otros 2 casos son parecidos, donde tags será un boolean opcional y enviroments un string obligatorio.

```

on:
  workflow_dispatch:
    inputs:
      alerta:
        description : 'Nivel'
        required : true
        default : medio
        type : choice
        option :
          - bajo
          - medio
          - alto

      tags:
        description : 'Opcional'
        required : false
        ****type : boolean

      enviroment:
        description : 'Objetivo'
        required : true
        type : string

```

## Schedule

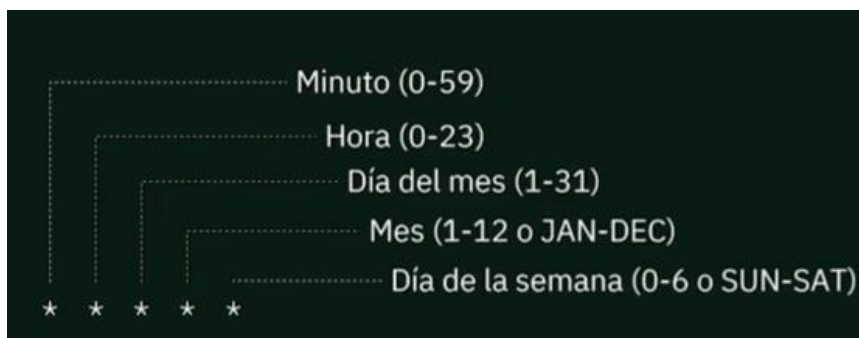
Los schedules son un tipo de trigger que se activará automáticamente por una regla de tiempo

```

on:
  schedule:
    - cron : '30 5,17 * * *'

```

Puedes notar que hay 2 valores puestos (30 - 5,17) y 3 asteriscos, esta es una notación de cronograma que indicará valores a nivel de tiempo.



Si el valor tiene un asterisco significa que será cíclico, si tiene coma significa que agrupará 2 valores

Por lo que nuestro '30 5,17 \* \* \*' significa a las 5:30AM y 5:30PM todos los días todos los meses del año.



## Uso de triggers y expresiones

Los crons corren en el horario UTC, para correrlo en el horario que deseamos, debemos investigar la semejanza con nuestro horario.

Literales, son valores que se pueden asignar a variables u otros

```
env:
  nulo: ${ { null } }
  booleano: ${ { false } }
  numeroEntero: ${ { 711 } }
  numeroFlotante: ${ { -9.2 } }
  numeroHexadecimal: ${ { 0xff } }
  numeroExponencial: ${ { -2.99e-2 } }
  string: Hola
  strinComillas: ${ { 'Hola' } }
```

### Operadores

Operador	Descripción
()	Agrupación lógica
[]	Índice
.	Referencia de propiedad
!	Not
<	Menor que
<=	Menor o igual que
>	Mayor que
>=	Mayor o igual que
==	Igual
!=	No igual a
&&	Y
	Or

### Funciones

Para ver mas funciones de Github Action consultar la siguiente URL

<https://docs.github.com/en/actions/reference/workflows-and-actions/expressions>

#### **if:**

if: \${ { <expresión> } }

#### **contains:**

contains(github.event.issue.labels.\*.name, 'bug')

#### **toJSON:**

toJSON(value)

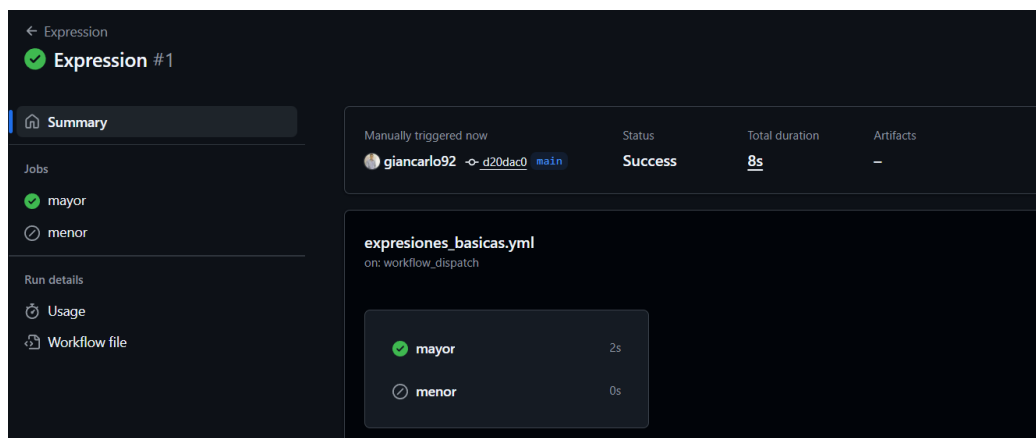
#### **fromJSON:**

fromJSON(value)

Los eventos del tipo `workflow_dispatch`, son aquellos que corren solo de forma manual, es decir dándole ejecutar en la UI de Github Actions.

```
name: Expression
on:
  workflow_dispatch:
    inputs:
      edad:
        description: 'Edad'
        required: true
        type: integer
      nombre:
        description: 'Tu nombre'
        required: true
        default: 'Juan'
        type: string
jobs:
  mayor:
    if: ${{ inputs.edad >= 18 }}
    runs-on: ubuntu-latest
    steps:
      - name: Correr script
        run: echo ${{ inputs.nombre }} es mayor de edad
  menor:
    if: ${{ inputs.edad < 18 }}
    runs-on: ubuntu-latest
    steps:
      - name: Correr script
        run: echo ${{ inputs.nombre }} es menor de edad
```

Despues de la configuración en Actions, se procede a correr el yaml



The screenshot shows the GitHub Actions interface for a workflow named "Expression". The workflow is marked as "Success" and was manually triggered by user "giancarlo92" on the "main" branch. The total duration of the run is 8 seconds. The interface displays a summary of the jobs: "mayor" (2s) and "menor" (0s). The workflow file is named "expresiones\_basicas.yml" and is triggered by the "workflow\_dispatch" event.

Manually triggered now	Status	Total duration	Artifacts
giancarlo92  d20dac0 <a href="#">main</a>	Success	8s	-

**expresiones\_basicas.yml**  
on: workflow\_dispatch

Job	Duration
mayor	2s
menor	0s

Reto propuesto

```
name: Expresiones Avanzadas

on:
  workflow_dispatch:
    inputs:
      usuario:
        description: 'Nombre de usuario'
        required: true
        type: string
      pais:
        description: 'País de origen'
        required: true
        type: string
      permiso_especial:
        description: '¿Tiene permiso especial?'
        required: false
```

```
    type: boolean
    default: false

env:
  SALUDO: "Hola"
  PAISES_PERMITIDOS: ["España", "Mexico", "Colombia"]

jobs:
  validar_y_saludar:
    runs-on: ubuntu-latest
    steps:
      - name: Validar país
        run: |
          if [[ "${{ contains(fromJSON(env.PAISES_PERMITIDOS), inputs.pais) }}" == "true" ]];
        then
          echo "El país '${{ inputs.pais }}' está en la lista de permitidos."
          echo "pais_valido=true" >> $GITHUB_ENV
        else
          echo "El país '${{ inputs.pais }}' no está en la lista permitida."
          echo "pais_valido=false" >> $GITHUB_ENV
        fi

      - name: Saludar al usuario
        if: env.pais_valido == 'true'
        run: |
          echo "${{ env.SALUDO }} ${{ inputs.usuario }}"
          echo "::notice title=Saludo::${{ env.SALUDO }} ${{ inputs.usuario }}"

      - name: Mostrar permiso especial
        if: ${{ inputs.permiso_especial }}
        run: |
          echo "El usuario ${{ inputs.usuario }} tiene permiso especial."
          echo "::notice title=Permiso Especial::El usuario ${{ inputs.usuario }} tiene permiso especial."

      - name: Mensaje de país no permitido
        if: env.pais_valido == 'false'
        run: |
          echo "El país '${{ inputs.pais }}' no está en la lista permitida. No se puede continuar."
          echo "::notice title=Error de País::El país '${{ inputs.pais }}' no está en la lista permitida. No se puede continuar."
```

