

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.

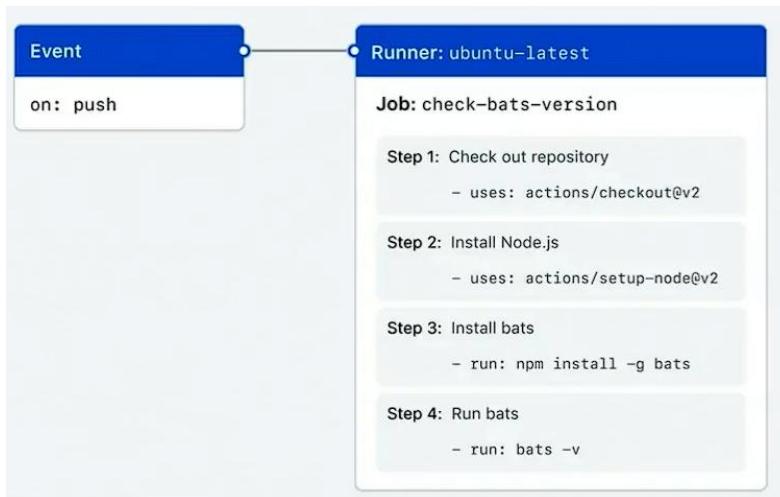


Step

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

The screenshot shows the GitHub Marketplace interface. At the top, there's a search bar with placeholder text 'Search for apps and actions' and a dropdown menu set to 'Sort: Best Match'. Below the search bar, there are three tabs: 'Types' (highlighted), 'Apps', and 'Actions' (selected). A sidebar on the left lists 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, and Utilities. The main area is titled 'Actions' and contains the sub-header 'An entirely new way to automate your development workflow.' It shows 17472 results filtered by 'Actions'. Each result card includes a play icon, the action name, the number of actions it contains, a brief description, and the number of stars. For example, 'First interaction' has 575 stars, 'Close Stale Issues' has 910 stars, 'Setup .NET Core SDK' has 714 stars, 'Setup Java JDK' has 1k stars, 'Download a Build Artifact' has 891 stars, 'Upload a Build Artifact' has 2.2k stars, 'Setup Go environment' has 1k stars, 'Setup Node.js environment' has 2.8k stars, and 'Cache' has 3.5k stars. There's also a card for 'ServiceNow DevOps Register Artifact' by ServiceNow.

Category	Action Name	Actions	Stars
API management	First interaction	By actions	575 stars
Chat	Close Stale Issues	By actions	910 stars
Code quality	Setup .NET Core SDK	By actions	714 stars
Code review	Setup Java JDK	By actions	1k stars
Continuous integration	Download a Build Artifact	By actions	891 stars
Dependency management	Upload a Build Artifact	By actions	2.2k stars
Deployment	Setup Go environment	By actions	1k stars
IDEs	Setup Node.js environment	By actions	2.8k stars
Learning	Cache	By actions	3.5k stars
Localization	ServiceNow DevOps Register Artifact	By ServiceNow	Not explicitly listed, but visible in the UI.
Mobile			
Monitoring			
Project management			
Publishing			
Recently added			
Security			
Support			
Testing			
Utilities			

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

The screenshot shows the GitHub Actions interface for the repository 'giancarlo92 / curso-github-action'. The 'Actions' tab is selected. On the left, there's a sidebar with a red arrow pointing to the 'New workflow' button. The main area shows a single workflow run for the 'Initial commit' with a green checkmark. The run details show it was triggered by a push and pushed by 'giancarlo92'.

Plantilla básica

The screenshot shows the 'Choose a workflow' page on GitHub. It features a search bar and a section titled 'Suggested for this repository' with a red arrow pointing to the 'Simple workflow' card. Below this, there's a 'Deployment' section with four cards: 'Deploy Node.js to Azure Web App', 'Deploy to Amazon ECS', 'Build and Deploy to GKE', and 'Terraform'. Each card has a 'Configure' button.

Esqueleto y acciones hechas por la comunidad

The screenshot shows the GitHub Actions workflow editor. On the left, a code editor displays a blank workflow file (blank.yml) with comments explaining the structure. On the right, a sidebar titled 'Marketplace' lists several actions: 'Cache' (5k stars), 'Setup Node.js environment' (4.3k stars), 'Download a Build Artifact' (1.6k stars), and 'Close Stale Issues' (1.5k stars). The GitHub logo is visible at the top.

```
1  # This is a basic workflow to help you get started with Actions
2
3  name: CI
4
5  # Controls when the workflow will run
6  on:
7    # Triggers the workflow on push or pull request events but only for the "main" branch
8    push:
9      branches: [ "main" ]
10     pull_request:
11       branches: [ "main" ]
12
13   # Allows you to run this workflow manually from the Actions tab
14   workflow_dispatch:
15
16   # A workflow run is made up of one or more jobs that can run sequentially or in parallel
17   jobs:
18     # This workflow contains a single job called "build"
19     build:
20       # The type of runner that the job will run on
21       runs-on: ubuntu-latest
22
```

Hola mundo en Python

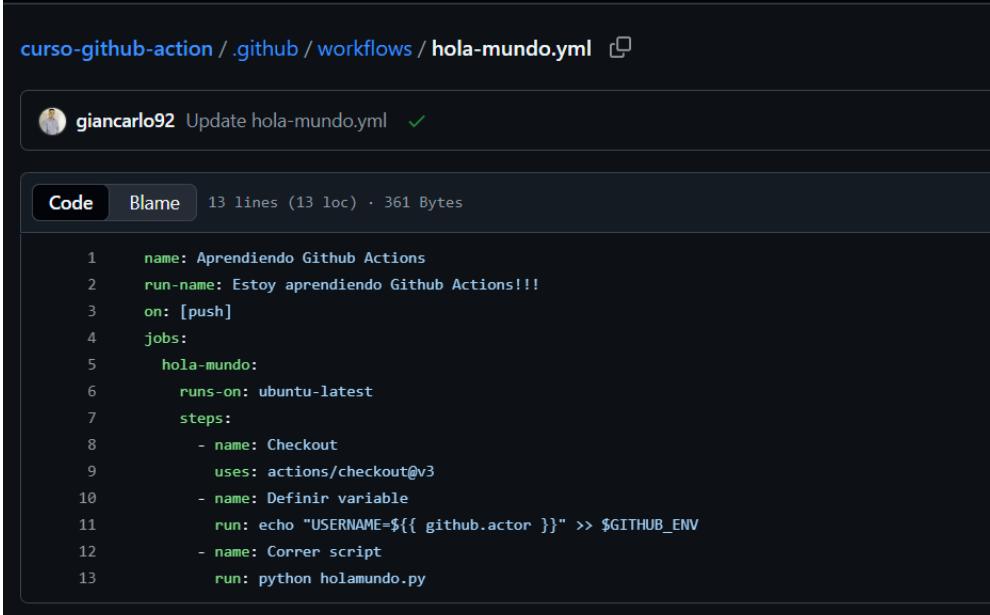
Archivo py

The screenshot shows a GitHub repository page for 'holamundo.py'. It displays the file content, which is a simple Python script that prints a greeting message using the 'os.getenv' function to get the 'USERNAME' environment variable. The file has 8 lines, 6 LOC, and 149 Bytes.

```
1 import os
2
3 def main():
4     nombre = os.getenv('USERNAME', 'usuario')
5     print(f"Hola {nombre} desde GitHub")
6
7 if __name__ == "__main__":
8     main()
```

Archivo yml (verificar la ruta)

Todos estos Jobs corren en el mismo espacio (servidor)



The screenshot shows a GitHub repository page for 'hola-mundo.yml'. It displays the workflow configuration, which defines a job named 'hola-mundo' that runs on 'ubuntu-latest'. The job consists of several steps: 'Checkout' (using 'actions/checkout@v3'), 'Definir variable' (running a command to set the USERNAME environment variable), and 'Correr script' (running 'python holamundo.py'). The workflow has 13 lines, 13 LOC, and 361 Bytes.

```
1 name: Aprendiendo Github Actions
2 run-name: Estoy aprendiendo Github Actions!!!
3 on: [push]
4 jobs:
5   hola-mundo:
6     runs-on: ubuntu-latest
7     steps:
8       - name: Checkout
9         uses: actions/checkout@v3
10        - name: Definir variable
11          run: echo "USERNAME=${{ github.actor }}" >> $GITHUB_ENV
12        - name: Correr script
13          run: python holamundo.py
```

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 comentario

```
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 environment, 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 optional y environments 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

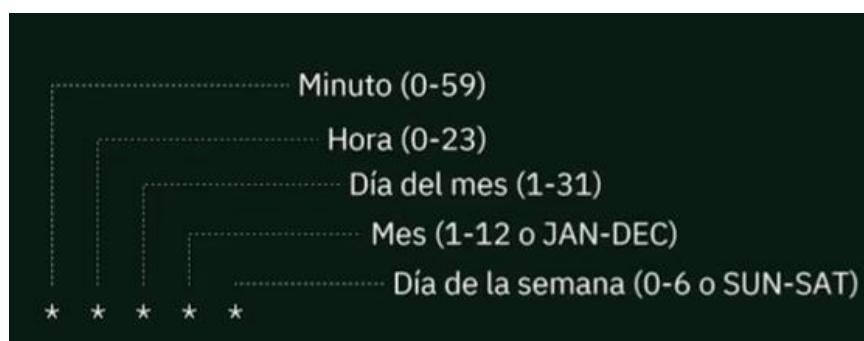
Los schedules son un tipo de trigger que se activará automáticamente por una regla de tiempo

```

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 cronos 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)
```

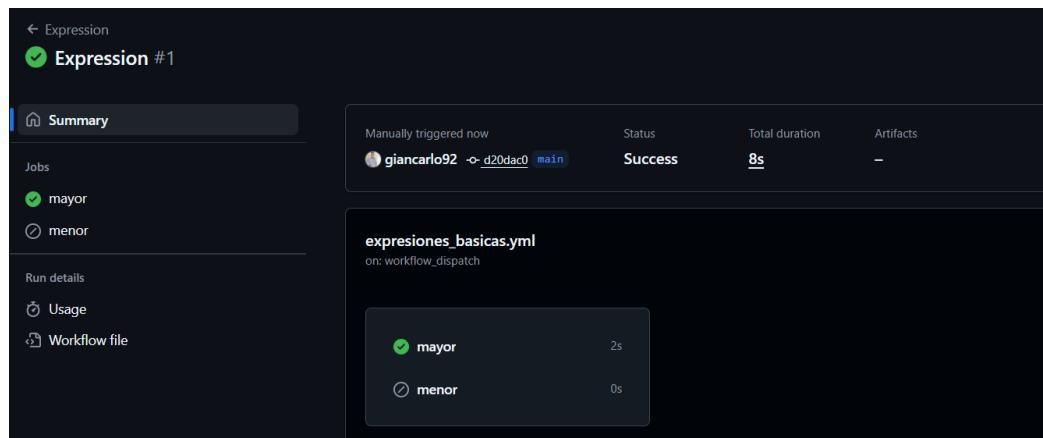
fromJSON:

```
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



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."

```

