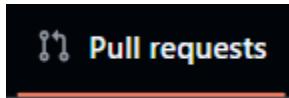


Guia paso a paso: GitHub Actions

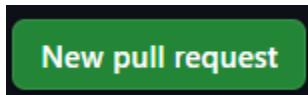
Hello GitHub Actions: <https://github.com/JCARL0/skills-hello-github-actions>

Parte 1-

Ve a la pestaña Pull Requests del repositorio..



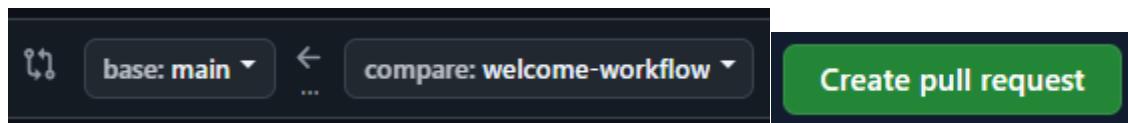
Haz clic en "New pull request".



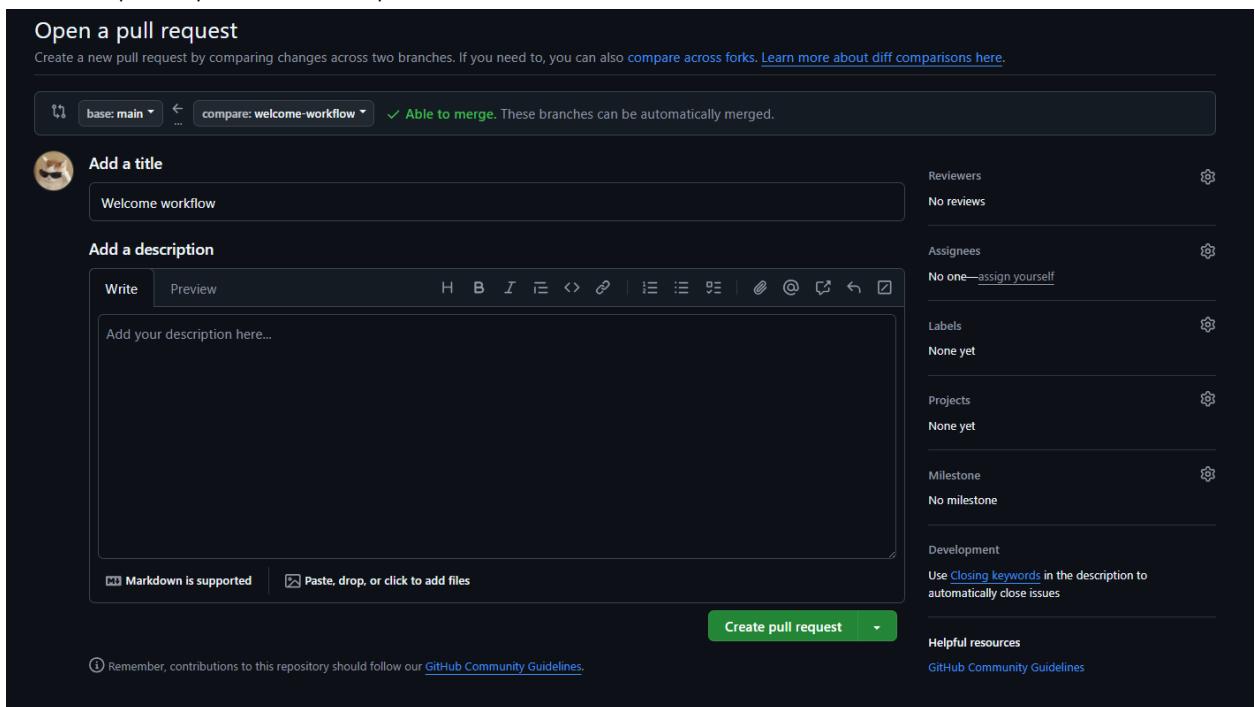
Crea un nuevo pull request desde la rama welcome-workflow hacia la rama main.

Base: main

Compare: welcome-workflow

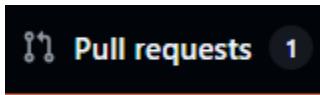


Dale click al create pull requests dos veces para confirmar los cambios

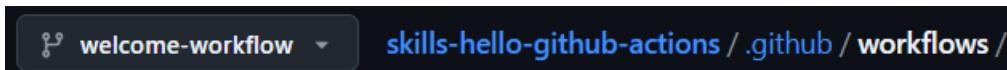


Parte 2-

Volvemos al pull request



Cambia la rama al welcome-workflow y navegamos en la carpeta siguiente

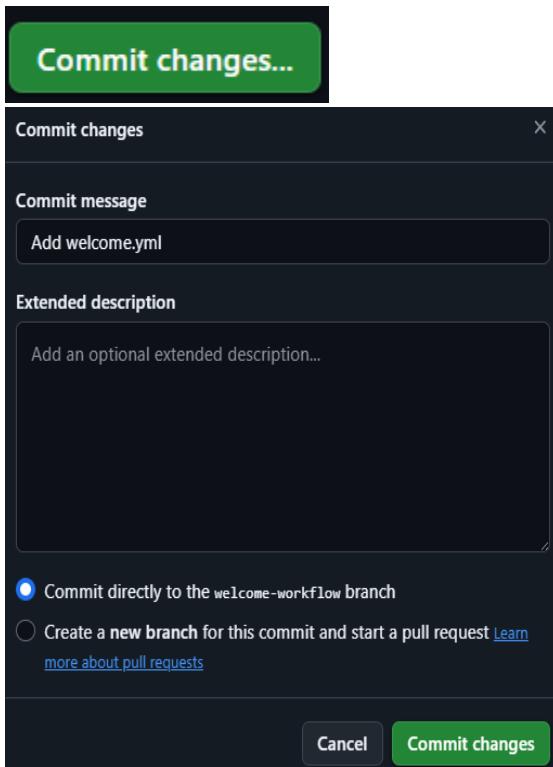


Creamos un nuevo fichero llamado welcome.yaml con el siguiente contenido

```
1 name: Post welcome comment
2 on:
3   pull_request:
4     types: [opened]
5   permissions:
6     pull-requests: write
```

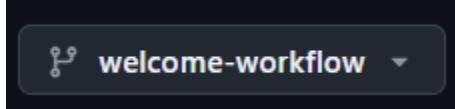
A screenshot of the GitHub 'Add file' dialog. It shows a button labeled 'Add file' and a text input field containing 'welcome.yaml'. Below the input field is a code editor displaying the YAML configuration for a GitHub Action.

Dale commit directo a la rama welcome-workflow

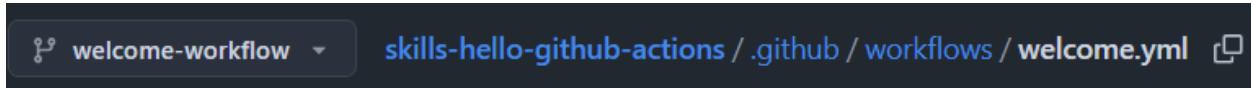


PT2

Asegura que la rama este en welcome-workflow



Editaremos el fichero welcome.yaml con siguiente contenido

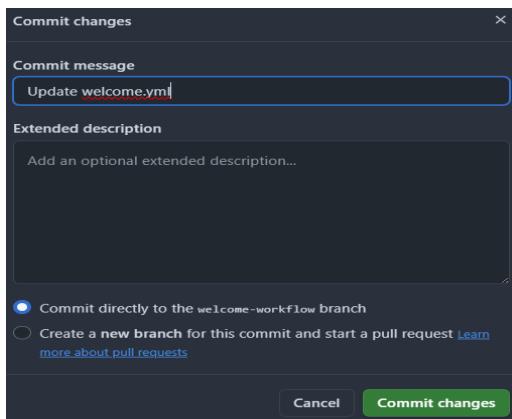


Icono para editar el fichero



```
1   name: Post welcome comment
2   on:
3     pull_request:
4       types: [opened]
5   permissions:
6     pull-requests: write
7   jobs:
8     build:
9       name: Post welcome comment
10      runs-on: ubuntu-latest
```

Una vez editada confirmamos los cambios



PT3

PT3

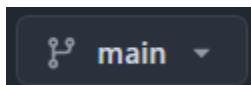
Vuelve editar el fichero con los cambios siguientes

The screenshot shows the GitHub Actions workflow editor for the repository 'skills-hello-github-actions'. The workflow file 'welcome.yml' is displayed, containing the following YAML code:

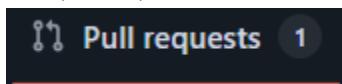
```
1 name: Post welcome comment
2 on:
3   pull_request:
4     types: [opened]
5   permissions:
6     pull-requests: write
7   jobs:
8     build:
9       name: Post welcome comment
10      runs-on: ubuntu-latest
11      steps:
12        - run: gh pr comment $PR_URL --body "Welcome to the repository!"
13        env:
14          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
15          PR_URL: ${{ github.event.pull_request.html_url }}
```

PT4

Hacemos merge desde la rama main



Ve a pull requests



Selecciona Welcome Workflow

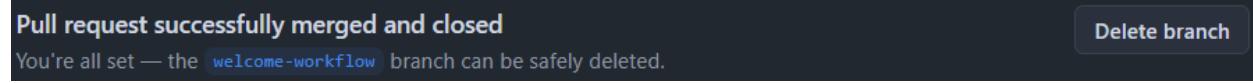
The screenshot shows the GitHub pull request merge dialog for pull request #1. The dialog includes fields for 'Commit message' (Merge pull request #1 from JCARLO/welcome-workflow) and 'Extended description' (Welcome workflow). At the bottom, it shows the author information (This commit will be authored by 147548026+JCARLO@users.noreply.github.com) and two buttons: 'Confirm merge' and 'Cancel'.

Resultado final despues de meargear

Pull request successfully merged and closed

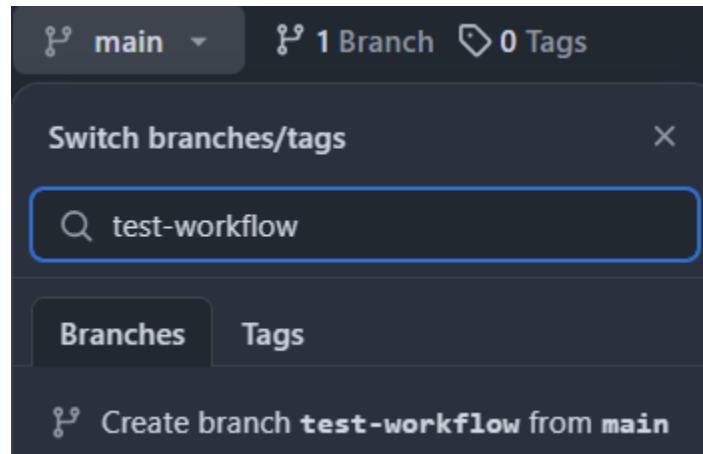
You're all set — the `welcome-workflow` branch can be safely deleted.

Delete branch



PT5

Creamos una nueva rama llamado test-workflow



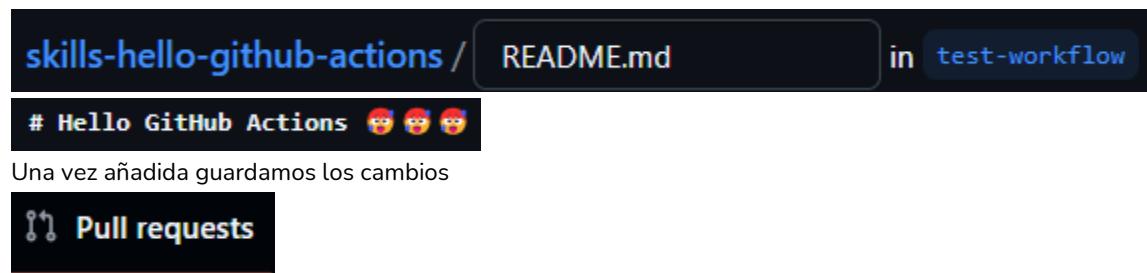
Switch branches/tags

test-workflow

Branches Tags

Create branch `test-workflow` from `main`

Editaremos el readme añadiendo emoticonos



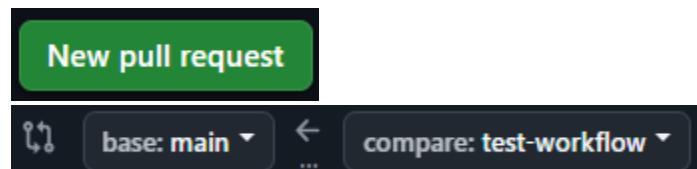
skills-hello-github-actions / README.md in test-workflow

Hello GitHub Actions 😊😊😊

Una vez añadida guardamos los cambios

Pull requests

Añadimos una nueva pull request



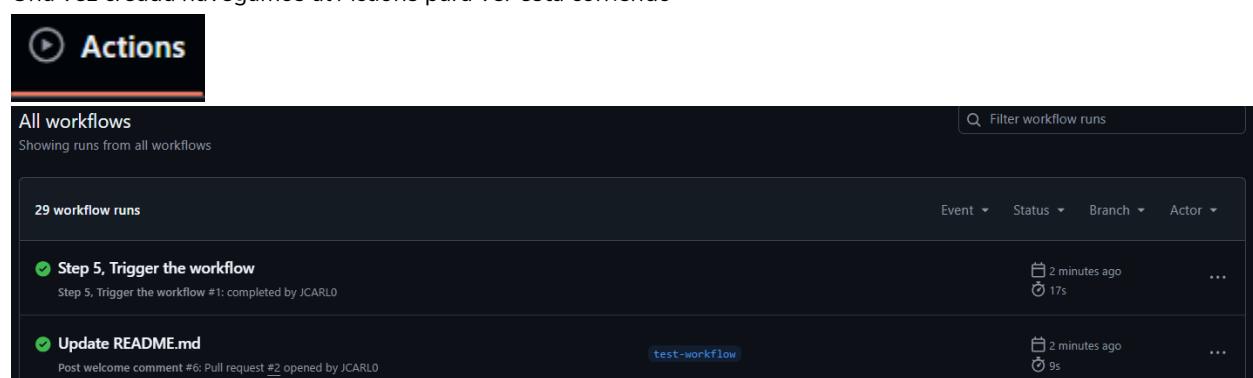
New pull request

base: main compare: test-workflow

Base: main

compare: test-workflow

Una vez creada navegamos al Actions para ver esta corriendo



All workflows

29 workflow runs

Event Status Branch Actor

Step 5, Trigger the workflow Step 5. Trigger the workflow #1: completed by JCARLO 2 minutes ago 17s ...

Update README.md Post welcome comment #6: Pull request #2 opened by JCARLO test-workflow 2 minutes ago 9s ...

FIN

Hello GitHub Actions

Create and run a GitHub Actions workflow.

Finish

Congratulations friend, you've completed this course!

Here's a recap of all the tasks you've accomplished in your repository:

- You've created your first GitHub Actions workflow file.
- You learned where to make your workflow file.
- You defined an event trigger, a job, and a step for your workflow.
- You're ready to automate anything you can dream of.

What's next?

- Learn more about GitHub Actions by reading "[Learn GitHub Actions](#)"
- Use actions created by others in [awesome-actions](#)
- We'd love to hear what you thought of this course [in our discussion board](#)
- [Take another course on GitHub Actions](#)
- Learn more about GitHub by reading the "[Get started](#)" docs
- To find projects to contribute to, check out [GitHub Explore](#)



Test with Actions

PT1

Añadimos un nuevo test



Dale new workflow y buscamos Simple Workflow

The screenshot shows the GitHub Actions interface for creating a new workflow. A search bar at the top contains the text "Simple workflow". Below it, a sidebar lists categories: Deployment, Security, Continuous integration, Automation, and Pages. The main area displays two workflow options: "Simple workflow" by GitHub and "Manual workflow" by GitHub Actions. Both cards include a "Configure" button. The "Simple workflow" card has a note: "Start with a file with the minimum necessary structure." The "Manual workflow" card notes: "Simple workflow that is manually triggered." The "Automation" category is highlighted with a red dot next to its name.

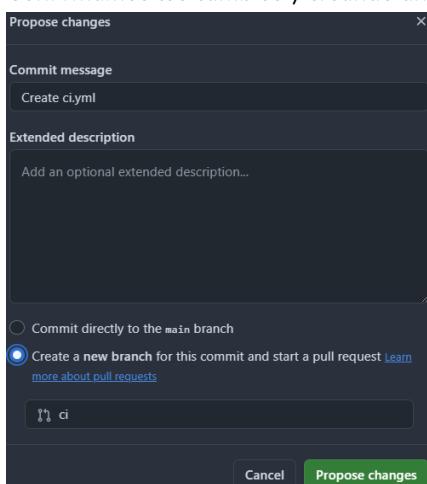
En .github/workflows creamos un nuevo fichero workflow llamado ci.yml



Añade el siguiente contenido al fichero

```
run: |
  npm install remark-cli remark-preset-lint-consistent
  npx remark . --use remark-preset-lint-consistent --frail
```

Confirmamos los cambios y creando una nueva rama llamado ci



PT2

Arreglamos el workflow

The screenshot shows the GitHub Actions interface. At the top, there's a header with "Step 2, Fix the test" and "2-fix-the-test.yml". Below it, a table shows "1 workflow run". A tooltip indicates the workflow has a "workflow_dispatch" event trigger. A specific run is highlighted: "Step 2, Fix the test #1: Manually run by JCARLO". The run status is "Success" with a duration of "19s". A "Run workflow" button is visible. The main area displays the workflow details for "2-fix-the-test.yml", which runs on "workflow_dispatch". It shows two steps: "Check current step number" and "On fix test", both of which were successful.

PT3

Test error

The screenshot shows the GitHub Actions CI pipeline for the repository "skills-test-with-actions/.github/workflows/ci.yml". A pull request from "JCARLO" is shown, with the commit message "Update ci.yml". The "Code" tab is selected, displaying the workflow file content:

```
1   build:
2     runs-on: ubuntu-latest
3     steps:
4       - uses: actions/checkout@v4
5
6       - name: Run markdown lint
7         run: |
8           npm install remark-cli remark-preset-lint-consistent vfile-reporter-json
9           npx remark . --use remark-preset-lint-consistent --report vfile-reporter-json 2> remark-lint-report.json
10
11      - uses: actions/upload-artifact@v4
12        with:
13          name: remark-lint-report
14          path: remark-lint-report.json
```

PT4

Añadimos una regla en las ramas

The screenshot shows the GitHub repository settings interface. The left sidebar has sections for General, Access, Code and automation, Security, and Integrations. Under 'Code and automation', 'Branches' is selected, which is also highlighted in blue in the main content area. The main content area is titled 'Branch protection rules'. It displays a message: 'Classic branch protections have not been configured' with a link to learn more about repository rules and protected branches. Below this are two buttons: 'Add branch ruleset' and 'Add classic branch protection rule'. The 'Add classic branch protection rule' button is highlighted with a red box.

Haz click al "add classic branch protection rule"

This is a modal window titled 'Classic branch protections have not been configured'. It contains a brief description: 'Define branch rules to disable force pushing, prevent branches from being deleted, or require pull requests before merging.' Below this are two buttons: 'Add branch ruleset' and 'Add classic branch protection rule'. The 'Add classic branch protection rule' button is highlighted with a red box.

Añade el nombre de la rama principal

This is a modal window with a single input field labeled 'Branch name pattern *'. The input field contains the value 'main'. The entire modal window is highlighted with a red box.

- Require a pull request before merging
- X Require approvals
- ✓ Require status checks to pass before merging

Protect matching branches

Require a pull request before merging
When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.

- Require approvals**
When enabled, pull requests targeting a matching branch require a number of approvals and no changes requested before they can be merged.
- Dismiss stale pull request approvals when new commits are pushed**
New reviewable commits pushed to a matching branch will dismiss pull request review approvals.
- Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.
- Require approval of the most recent reviewable push**
Whether the most recent reviewable push must be approved by someone other than the person who pushed it.

Require status checks to pass before merging
Choose which [status checks](#) must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

- Require branches to be up to date before merging**
This ensures pull requests targeting a matching branch have been tested with the latest code. This setting will not take effect unless at least one status check is enabled (see below).

No required checks

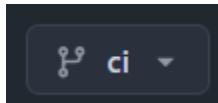
No checks have been added

Branch protection rules

[Add rule](#)

main	Currently applies to 1 branch	Edit	Delete
------	-------------------------------	----------------------	------------------------

Ve a la branch para guardar los cambios



PT5

1. Ve en Pull requests
2. Click Merge pull request

Pull requests 1

Create ci.yml
#1 opened 44 minutes ago by JCARLO

Merge pull request ▾

FIN

Test with Actions

Create workflows that enable you to use Continuous Integration (CI) for your projects.

Finish

Congratulations friend, you've completed this course!

Here's a recap of all the tasks you've accomplished in your repository:

- We created an Actions workflow to lint our Markdown files.
- You caught an issue in a file and fixed the issue before it could make it to `main`.
- You learned how to use build artifacts for test reports.
- You enabled branch protections to require the workflow to pass before merging.



What's next?

- Get more ideas of what you can do with [awesome actions](#).
- We'd love to hear what you thought of this course [in our discussion board](#).
- [Take another GitHub Skills course](#).
- [Read the GitHub Getting Started docs](#).
- To find projects to contribute to, check out [GitHub Explore](#).

Write JavaScript actions

PT1

Abre el terminal o command prompt

```
C:\Users\Jcarl\Documents\M8UF4PF>git clone https://github.com/JCARL0/skills-write-javascript-actions.git
Cloning into 'skills-write-javascript-actions'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 32 (delta 7), reused 13 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (32/32), 16.93 KiB | 1.21 MiB/s, done.
Resolving deltas: 100% (7/7), done.
```

```
C:\Users\Jcarl\Documents\M8UF4PF>cd skills-write-javascript-actions
```

Cambia el Branch a main

```
C:\Users\Jcarl\Documents\M8UF4PF\skills-write-javascript-actions>git switch main
Already on 'main'
Your branch is up to date with 'origin/main'.
```

Crear una nueva carpeta en github/actions/joke-action

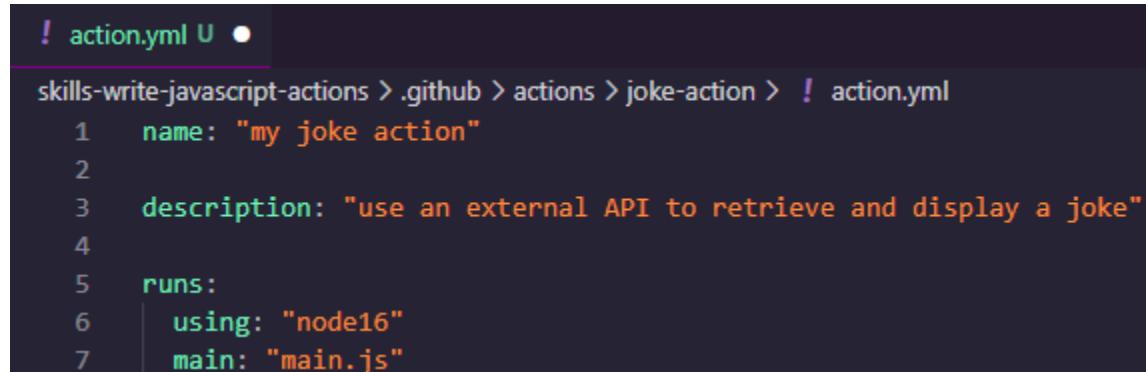
```
C:\Users\Jcarl\Documents\M8UF4PF\skills-write-javascript-actions>mkdir .github\actions\joke-action
npm init -y para inicializar el proyecto
```

Instalar request, request-promise and @actions/core dependencies usando npm

```
C:\Users\Jcarl\Documents\M8UF4PF\skills-write-javascript-actions\.github\actions\joke-action>npm install --save request request-promise @actions/core
hacemos commit para añadir los nuevos ficheros al nuestro repositorio
C:\Users\Jcarl\Documents\M8UF4PF\skills-write-javascript-actions\.github\actions\joke-action>git add .
C:\Users\Jcarl\Documents\M8UF4PF\skills-write-javascript-actions>git commit -m "add project dependencies"
C:\Users\Jcarl\Documents\M8UF4PF\skills-write-javascript-actions>git push
```

PT2

Creacion de un nuevo fichero en ".github/actions/joke-action/action.yml"



```
! action.yml U ●
skills-write-javascript-actions > .github > actions > joke-action > ! action.yml
1   name: "my joke action"
2
3   description: "use an external API to retrieve and display a joke"
4
5   runs:
6     using: "node16"
7     main: "main.js"
```

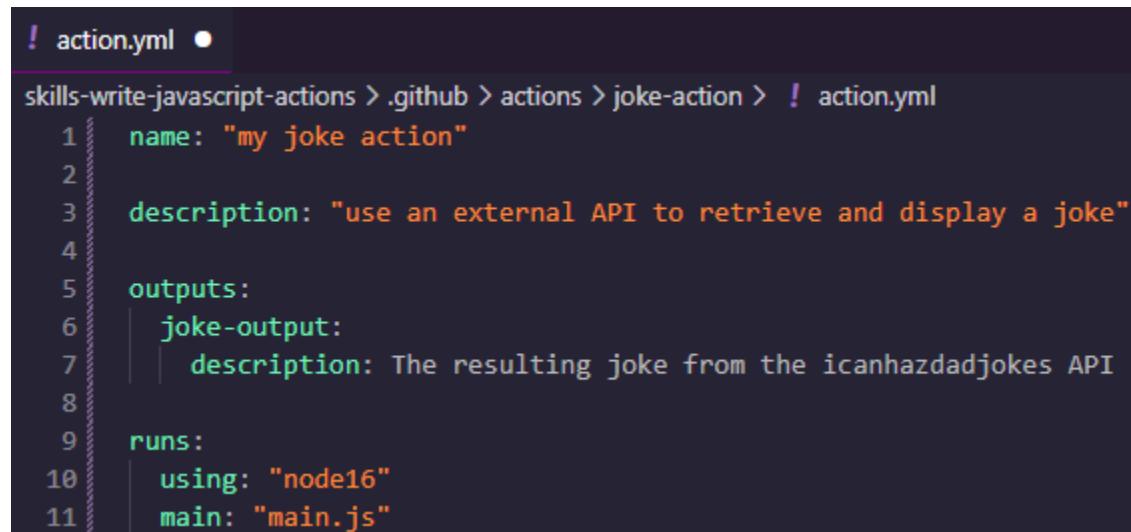
Confirme los cambios

```
PS C:\Users\Jcarl\Documents\M8UF4PF\skills-write-javascript-actions\.github\actions\joke-action> git add action.yml
PS C:\Users\Jcarl\Documents\M8UF4PF\skills-write-javascript-actions\.github\actions\joke-action> git commit -m 'create action.yml'
[main 0327cf5] create action.yml
 1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 .github/actions/joke-action/action.yml
PS C:\Users\Jcarl\Documents\M8UF4PF\skills-write-javascript-actions\.github\actions\joke-action> git pull
```

```
PS C:\Users\Jcarl\Documents\M8UF4PF\skills-write-javascript-actions\.github\actions\joke-action> git push
fatal: unable to connect to cache daemon: Unknown error
Enumerating objects: 17, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 1.04 KiB | 353.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/JCARL0/skills-write-javascript-actions.git
  8f285c3..5df0770  main -> main
```

PT3

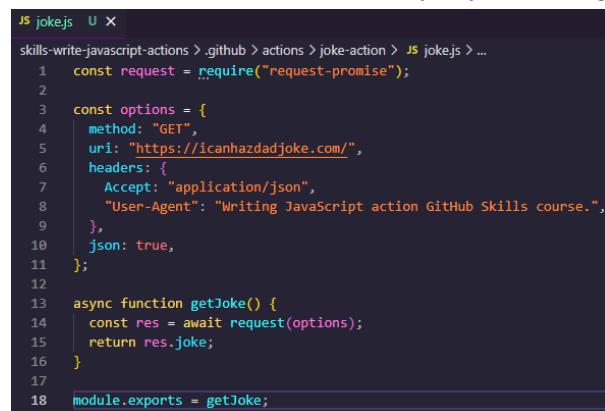
Edita el ".github/actions/joke-action/action.yml" y guarda los cambios



```
! action.yml
skills-write-javascript-actions > .github > actions > joke-action > ! action.yml
1   name: "my joke action"
2
3   description: "use an external API to retrieve and display a joke"
4
5   outputs:
6     joke-output:
7       description: The resulting joke from the icanhazdadjokes API
8
9   runs:
10    using: "node16"
11    main: "main.js"
```

PT4

Creacion de un nuevo fichero llamado joke.js con los siguientes contenidos



```
joke.js  U X
skills-write-javascript-actions > .github > actions > joke-action > joke.js > ...
1  const request = require("request-promise");
2
3  const options = {
4    method: "GET",
5    uri: "https://icanhazdadjoke.com/",
6    headers: {
7      Accept: "application/json",
8      "User-Agent": "Writing JavaScript action GitHub Skills course.",
9    },
10   json: true,
11 };
12
13 async function getJoke() {
14   const res = await request(options);
15   return res.joke;
16 }
17
18 module.exports = getJoke;
```

Creacion un nuevo fichero llamado main.js con los siguientes contenidos

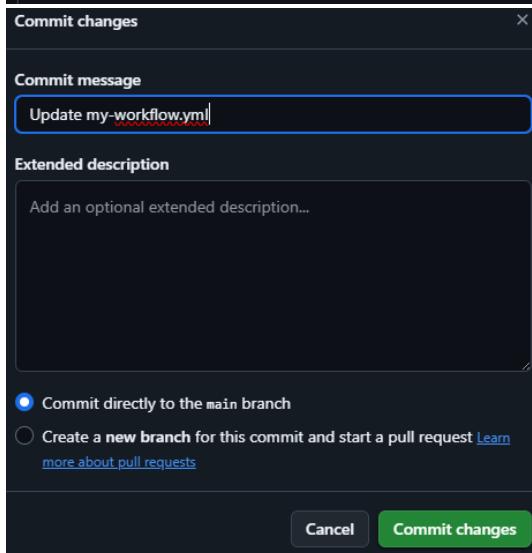
```
JS main.js U X  
skills-write-javascript-actions > .github > actions > joke-action > JS main.js > ...  
1 const getJoke = require("./joke");  
2 const core = require("@actions/core");  
3  
4 √ async function run() {  
5   const joke = await getJoke();  
6   console.log(joke);  
7   core.setOutput("joke-output", joke);  
8 }  
9  
10 run();
```

PT5

Edita el fichero "my-workflow.yml" y hacer commit directamente al Branch main

The screenshot shows the GitHub Actions workflow editor for the repository "skills-write-javascript-actions". The workflow file "my-workflow.yml" is open, showing the following YAML configuration:

```
name: JS Actions  
on:  
  issues:  
    types: [labeled]  
jobs:  
  action:  
    if: ${{ !github.event.repository.is_template }}  
    runs-on: ubuntu-latest  
    steps:  
      - uses: actions/checkout@v4  
      - name: ha-ha  
        uses: ./github/actions/joke-action
```



PT6

Trigger joke action

A screenshot of a GitHub Issues page. It shows two items in the list:

- issue #2 second joke
#3 · JCARL0 opened now
- issue #1 first joke
#2 · JCARL0 opened 1 minute ago

Write JavaScript Actions

Write your own GitHub JavaScript Action and automate customized tasks unique to your workflow.

Finish

Congratulations, you've completed this course!

In this course, you've learned a lot about developing custom actions using JavaScript and Actions Toolkit.



Publishing your actions

Publishing your actions is a great way to help others in your team and across the GitHub community. Although actions do not need to be published to be consumed, by adding them to the marketplace you make them easier to find.

Some notable actions you will find on the marketplace are:

- [Actions for Discord](#)
- [GitHub Action for Slack](#)
- [Jekyll action](#)
- [Run Jest](#)

And that just scratches the surface of the 1600+ and counting actions you will find on the marketplace

Follow [this guide](#) to learn how to publish your actions to the GitHub Marketplace

What's next?

- [Take another GitHub Skills course.](#)
- We'd love to hear what you thought of this course in our [discussion board](#).
- [Read the GitHub Getting Started docs.](#)
- To find projects to contribute to, check out [GitHub Explore](#).