

1 - Pros y Contras

2- Configurando GitHub Actions

The screenshot shows the GitHub Actions 'new' page for the repository `Manuel-Bobadilla/spring-boot`. The page has a dark theme. At the top, there's a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below this, the repository name and 'Public' status are shown. A secondary navigation bar includes links for Code, Issues, Pull requests, Actions (which is highlighted), Projects, Wiki, Security, Insights, and Settings. The main content area features a large heading 'CI/CD made fast, easy and free' followed by a paragraph explaining GitHub Actions. Below this is a link to 'Choose a workflow to get started'. A video player is embedded on the right side of the page. At the bottom, there's a 'Workflows' section with a search bar and a 'Suggested for this repository' link.

Activities Google Chrome sep 29 14:06

MI unidad - Google Drive TP 8 herramientas nube Ing-software-3/08-herre Actions - Manuel-Bobadilla Ing-software-3/8-trabaj

github.com/Manuel-Bobadilla/spring-boot/actions/new

Search or jump to... Pull requests Issues Marketplace Explore

Manuel-Bobadilla / spring-boot Public

Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

CI/CD made fast, easy and free

GitHub Actions is the simplest way to automate, scale, and secure your CI/CD pipeline. Leverage thousands of workflow templates or create your own. Use free minutes you already have, without needing to manage another tool. Save time, ship fast and build more... all within GitHub.

Choose a workflow to get started. Or skip this and set up a workflow yourself →

Workflows

Search workflows

Suggested for this repository

<https://github.com/Manuel-Bobadilla/spring-boot/pulse>

The screenshot shows the GitHub Actions 'new' page with the workflow editor open. The URL in the browser is `github.com/Manuel-Bobadilla/spring-boot/new/main?filename=.github%2Fworkflows%2Fmain.yml&workflow_template=blank`. The repository name is `Manuel-Bobadilla/spring-boot`. The editor shows a file named `main.yml` in the `main` branch. The code in the editor is a basic workflow template. On the right side, there's a 'Marketplace' section with a search bar and a list of 'Featured Actions' including 'Upload a Build Artifact', 'Setup Go environment', and 'Close Stale Issues'.

Activities Google Chrome sep 29 14:07

MI unidad - Google Drive TP 8 herramientas nube Ing-software-3/08-herre New File Ing-software-3/8-trabaj

github.com/Manuel-Bobadilla/spring-boot/new/main?filename=.github%2Fworkflows%2Fmain.yml&workflow_template=blank

Search or jump to... Pull requests Issues Marketplace Explore

Manuel-Bobadilla / spring-boot Public

Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

spring-boot / .github / workflows / main.yml in main

Cancel changes Start commit

Edit new file Preview Spaces 2 No wrap

```
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:
```

Marketplace Documentation

Search Marketplace for Actions

Featured Actions

- Upload a Build Artifact** By actions 1.8k
Upload a build artifact that can be used by subsequent workflow steps
- Setup Go environment** By actions 875
Setup a Go environment and add it to the PATH
- Close Stale Issues** By actions 757
Close issues and pull requests with no recent activity

<https://github.com/Manuel-Bobadilla/spring-boot/pulse>

Script

```
# This is a basic workflow to help you get started with Actions

name: CI

# Controls when the workflow will run
on:
  # Triggers the workflow on push or pull request events but only for the "main" branch
  push:
    paths:
      - '.*/**'
    branches: [ "main" ]
  pull_request:
    paths:
      - '.*/**'
    branches: [ "main" ]

# Allows you to run this workflow manually from the Actions tab
workflow_dispatch:

# A workflow run is made up of one or more jobs that can run sequentially or in parallel
jobs:
  # This workflow contains a single job called "build"
  build:
    # The type of runner that the job will run on
    runs-on: ubuntu-latest

    # Steps represent a sequence of tasks that will be executed as part of the job
    steps:
      # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
      - uses: actions/checkout@v3

      # Install Java JDK with maven
      - name: Set up JDK 8
        uses: actions/setup-java@v2
        with:
          java-version: '8'
          distribution: 'adopt'
          cache: maven

      # Compile the application
      - name: Build with Maven
        run: mvn -B package --file pom.xml
```

El pipeline define los eventos disparadores siendo los push y pull_request, luego se ejecuta un job donde se define en donde se va a correr la construcción (en este caso ubuntu), este job también está constituido por pasos, los cuales setean jdk 8, y luego corren el comando *mvn -B package --file pom.xml* para construir el proyecto

Corremos el build manualmente

The screenshot shows the GitHub Actions interface for the repository **Manuel-Bobadilla / spring-boot**. The **Actions** tab is selected, displaying a workflow named **CI #1** with a green status icon. A sidebar on the left shows the **Summary** and **Jobs** sections, with **build** listed under jobs. The main area shows the details of the **build** job, which succeeded 1 minute ago in 37s. A table lists the steps of the job:

Step	Duration
Set up job	2s
Run actions/checkout@v3	1s
Set up JDK 8	6s
Build with Maven	23s
Post Set up JDK 8	3s
Post Run actions/checkout@v3	0s
Complete job	0s

The URL at the bottom of the page is <https://github.com/Manuel-Bobadilla/spring-boot/pulse>.

3- Utilizando nuestros proyectos con Docker

Seteamos los secretos Usuario y Contraseña para que el workflow los tome desde las variables secretas del repositorio

Script para pushear a **docker hub**

```
name: Docker

on:
  workflow_dispatch:
  push:
    branches: [ $default-branch ]
    # Publish semver tags as releases.
    tags: [ 'v*.*.*' ]
  pull_request:
    branches: [ $default-branch ]

env:
  # Use docker.io for Docker Hub if empty
  REGISTRY: ghcr.io
  # github.repository as <account>/<repo>
  IMAGE_NAME: ${github.repository}

jobs:
  build:

    runs-on: ubuntu-latest
    permissions:
      contents: read
      packages: write
      # This is used to complete the identity challenge
      # with sigstore/fulcio when running outside of PRs.
      id-token: write

    steps:
      - name: Checkout repository
        uses: actions/checkout@v3

      # Workaround: https://github.com/docker/build-push-action/issues/461
      - name: Setup Docker buildx
        uses: docker/setup-buildx-action@79abd3f86f79a9d68a23c75a09a9a85889262adf

      # Login against a Docker hub
      - name: Log into docker hub
        if: github.event_name != 'pull_request'
        uses: docker/login-action@v2
        with:
          username: ${secrets.DOCKER_USER}
          password: ${secrets.DOCKER_PASSWORD}

      # Build and push Docker image with Buildx (don't push on PR)
      # https://github.com/docker/build-push-action
      - name: Build and push Docker image
        id: build-and-push
```

```

    uses: docker/build-push-action@v2
    with:
      context: .
      push: ${ github.event_name != 'pull_request' }
      tags: manuelbobadilla/spring-boot-github:latest

# Sign the resulting Docker image digest except on PRs.
# This will only write to the public Rekor transparency log when the Docker
# repository is public to avoid leaking data. If you would like to publish
# transparency data even for private images, pass --force to cosign below.
# https://github.com/sigstore/cosign
- name: Sign the published Docker image
  if: ${ github.event_name != 'pull_request' }
  env:
    COSIGN_EXPERIMENTAL: "true"
  # This step uses the identity token to provision an ephemeral certificate
  # against the sigstore community Fulcio instance.
  run: echo "${ steps.meta.outputs.tags }" | xargs -I {} cosign sign {}@${ steps.build-and-push.outputs.digest }

```

Script para pushear al servicio de contenedores de github

```

name: Docker2

# This workflow uses actions that are not certified by GitHub.
# They are provided by a third-party and are governed by
# separate terms of service, privacy policy, and support
# documentation.

on:
  workflow_dispatch:
  push:
    branches: [ $default-branch ]
    # Publish semver tags as releases.
    tags: [ 'v*.*.*' ]
  pull_request:
    branches: [ $default-branch ]

env:
  # Use docker.io for Docker Hub if empty
  REGISTRY: ghcr.io
  # github.repository as <account>/<repo>
  IMAGE_NAME: ${ github.repository }

jobs:
  build:

    runs-on: ubuntu-latest
    permissions:
      contents: read
      packages: write
      # This is used to complete the identity challenge
      # with sigstore/fulcio when running outside of PRs.
      id-token: write

```

steps:

```
- name: Checkout repository
  uses: actions/checkout@v3

# Install the cosign tool except on PR
# https://github.com/sigstore/cosign-installer
- name: Install cosign
  if: github.event_name != 'pull_request'
  uses: sigstore/cosign-installer@f3c664df7af409cb4873aa5068053ba9d61a57b6 #v2.6.0
  with:
    cosign-release: 'v1.11.0'

# Workaround: https://github.com/docker/build-push-action/issues/461
- name: Setup Docker buildx
  uses: docker/setup-buildx-action@79abd3f86f79a9d68a23c75a09a9a85889262adf

# Login against a Docker registry except on PR
# https://github.com/docker/login-action
- name: Log into registry ${ env.REGISTRY }
  if: github.event_name != 'pull_request'
  uses: docker/login-action@28218f9b04b4f3f62068d7b6ce6ca5b26e35336c
  with:
    registry: ${ env.REGISTRY }
    username: ${ github.actor }
    password: ${ secrets.GITHUB_TOKEN }

# Extract metadata (tags, labels) for Docker
# https://github.com/docker/metadata-action
- name: Extract Docker metadata
  id: meta
  uses: docker/metadata-action@98669ae865ea3cfffcbcaa878cf57c20bbf1c6c38
  with:
    images: ${ env.REGISTRY }/${ env.IMAGE_NAME }

# Build and push Docker image with Buildx (don't push on PR)
# https://github.com/docker/build-push-action
- name: Build and push Docker image
  id: build-and-push
  uses: docker/build-push-action@ac9327eae2b366085ac7f6a2d02df8aa8ead720a
  with:
    context: .
    push: ${ github.event_name != 'pull_request' }
    tags: ${ steps.meta.outputs.tags }
    labels: ${ steps.meta.outputs.labels }
    cache-from: type=gha
    cache-to: type=gha,mode=max

# Sign the resulting Docker image digest except on PRs.
# This will only write to the public Rekor transparency log when the Docker
# repository is public to avoid leaking data. If you would like to publish
# transparency data even for private images, pass --force to cosign below.
# https://github.com/sigstore/cosign
- name: Sign the published Docker image
```

```
if: ${ github.event_name != 'pull_request' }}
env:
  COSIGN_EXPERIMENTAL: "true"
  # This step uses the identity token to provision an ephemeral certificate
  # against the sigstore community Fulcio instance.
  run: echo "${ steps.meta.outputs.tags }}" | xargs -I {} cosign sign {}@${ steps.build-and-push.outputs.digest }}
```


