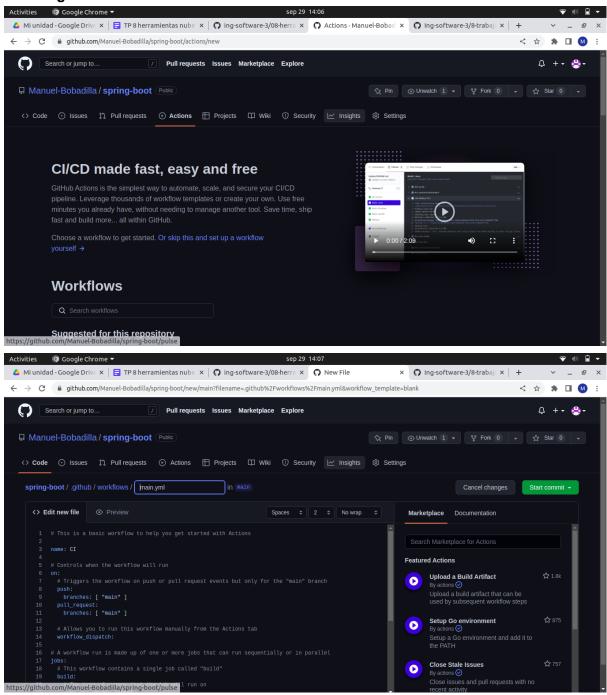
1 - Pros y Contras

2- Configurando GitHub Actions

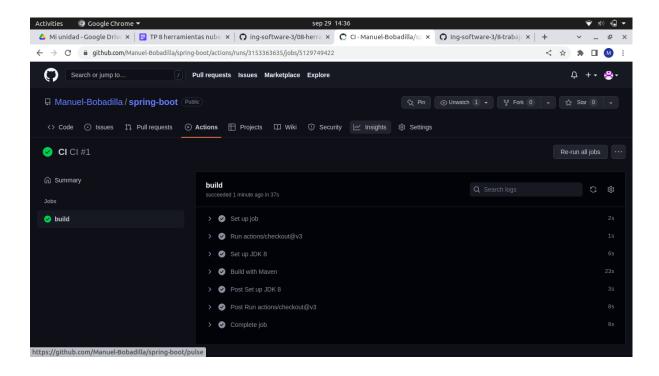


Script

```
# This is a basic workflow to help you get started with Actions
name: CI
# Controls when the workflow will run
 # 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
 # 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
     # 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 constituidos 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



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
 workflow_dispatch:
 push:
   branches: [ $default-branch ]
   # Publish semver tags as releases.
   tags: [ 'v*.*.*' ]
 pull request:
   branches: [ $default-branch ]
 # 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.
  workflow_dispatch:
  push:
    branches: [ $default-branch ]
   # Publish semver tags as releases.
   tags: [ 'v*.*.*' ]
  pull request:
    branches: [ $default-branch ]
  # 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@98669ae865ea3cffbcbaa878cf57c20bbf1c6c38
     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
```