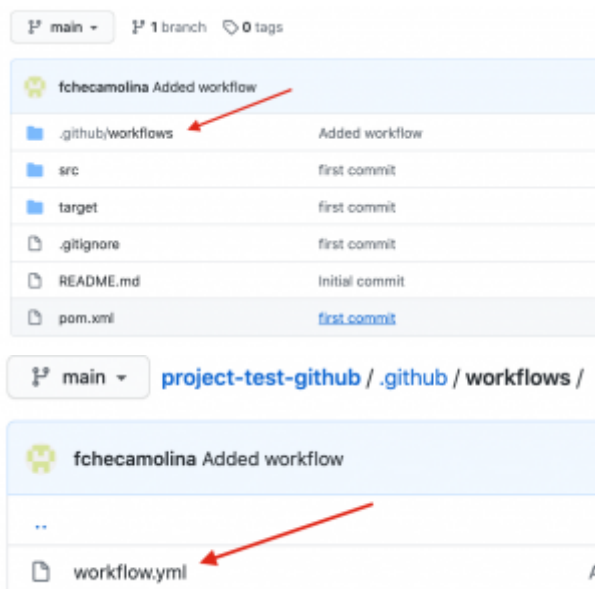


Para ver un ejemplo se puede tomar un ejemplo de un proyecto Java con Maven muy simple de este [enlace](#) o podéis utilizar cualquier proyecto que ya tengáis subido.

Lo primero de todo es crear la propia carpeta en la cual añadiremos el fichero de configuración del workflow. Esta carpeta tiene por norma que llamarse `.github/workflows/`. Aquí vamos a crear el fichero `workflow.yml` (Github Actions utiliza la sintaxis de [YAML](#) para especificar la configuración de todo el workflow).



El contenido inicial de nuestro **workflow.yml**

```
1name: Build and test of Java Project
2
3on: [push]
4
5jobs:
6  build:
7    runs-on: ubuntu-latest
8
9    steps:
10     - uses: actions/checkout@v2
11     - name: Set up JDK 1.8
12       uses: actions/setup-java@v1
```

```
13   with:
14     java-version: 1.8
15   - name: Build with Maven
16     run: mvn -B package --file pom.xml
```

Entendiendo el fichero por partes:

- **“name: Build and test of Java Project”**: El nombre opcional que le das al workflow
- **“on”**: Especifica el evento que automáticamente comienza a ejecutar el fichero de workflow. El ejemplo lo ejecuta gracias al comando **push** de git sobre nuestro repositorio. Para especificar además la rama o ramas sobre las que nos gustaría que iniciase, sería añadiendo:

```
1 on: [push]
2     Branches: [master]
```

- **«jobs»**: Sección donde se pueden especificar uno o más jobs.
- **«build»**: Es el nombre que le hemos dado a nuestro primer y único job. En este caso el nombre sí es **obligatorio**.
- **«runs on: ubuntu-latest»**: Configura el workflow para que se ejecute en una instancia de la última versión de ubuntu. Se puede cambiar por otro sistema operativo si quisiéramos: *windows-latest*, *macos-11.0*, etc. [Aquí](#) se pueden ver los disponibles.
- **«steps»**: Sección donde se especifican uno o más steps de un único job.
- **«uses: actions/checkout@v2»**: La palabra clave **uses** le dice al job de obtener **v2** (versión 2, antiguamente se usaba la v1) de la acción de la comunidad de Github llamada **actions/checkout**. Éste es un action que comprueba

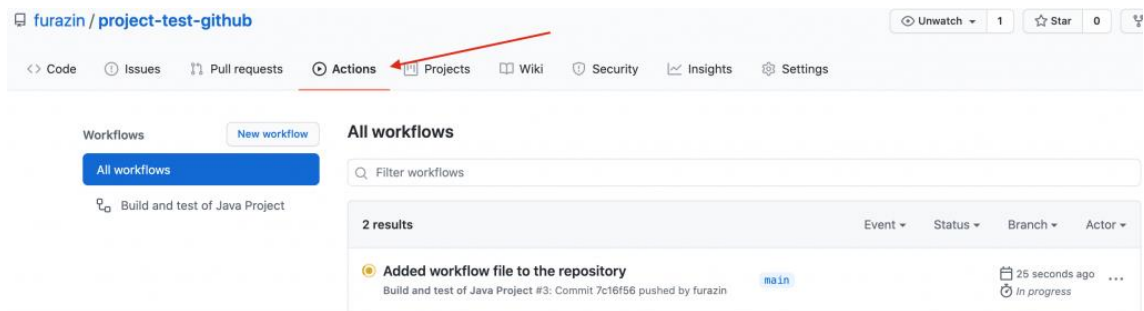
nuestro repositorio y lo descarga en nuestro runner o instancia, permitiendo que sobre el código podamos ejecutar el resto de acciones. Es **obligatorio** añadir este action de **checkout** las veces que nuestro workflow ejecute sobre nuestro código o se haga uso de un action que hemos definido en otro fichero del repositorio.

- «**name: Set up JDK 1.8**»: Un nombre opcional que se le ha dado al action.
- «**uses: actions/setup-java@v1**»: Este action se encarga de descargar e instalar una versión específica de java (**java-version: 1.8** como podemos ver) que la comunidad de Github ya ha preparado para poder utilizarse. [Aquí](#) se pueden consultar además todas las versiones disponibles y sintaxis adicional.
- «**run: mvn -B package -- file pom.xml**»: La palabra *run* le dice al job de ejecutar un comando en el runner. En este caso estamos utilizando **maven** para compilar y empaquetar nuestro proyecto.

Ahora para lanzar el workflow y ver los resultados, hacemos **push** de nuestro proyecto a la rama en la que lo tengamos (no importa la rama pues como no hemos especificado cuál, lo va a lanzar para todas) tras haber creado la carpeta y añadido el fichero de *workflow.yml*.

Hacemos git push

Justo después, nos dirigimos a la sección de **Actions** de nuestra página principal del repositorio en Github.



Como vemos, automáticamente se nos ha creado un proceso de **workflow** llamado «Added workflow file to the repository» y que se está ejecutando. El icono amarillo indica que aún no ha acabado.

Una vez termina satisfactoriamente, tiene que aparecer de la siguiente manera:



Y además se pueden consultar los logs de cada uno de los jobs que lo forman pulsando sobre él y luego seleccionando el job que queremos para ver si se ha ejecutado cada paso que hemos especificado.

Added workflow file to the repository

main 7c16f56

Build and test of Java Project
on: push

✓ build

build
succeeded 18 minutes ago in 24s

Build with Maven

```
534 [INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/s
1.15.jar (60 kB at 2.1 MB/s)
535 [INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/s
MB/s)
536 [INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/s
kB at 5.6 MB/s)
537 [INFO] Downloaded from central: https://repo.maven.apache.org/maven2/comm
MB/s)
538 [INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/s
4.3 MB/s)
539 [INFO] Building jar: /home/runner/work/project-test-github/project-test-g
540 [INFO] -----
541 [INFO] BUILD SUCCESS
542 [INFO] -----
543 [INFO] Total time: 13.642 s
544 [INFO] Finished at: 2020-10-27T12:57:32Z
545 [INFO] -----
```

> ✓ Post Set up JDK 1.8

> ✓ Post Run actions/checkout@v2

> ✓ Complete job

Además si en nuestro proyecto hemos añadido Tests, éstos se ejecutarán y se mostrarán sus resultados también:

```
490 [INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/
2.12.4.jar
491 [INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/s
2.12.4.jar (37 kB at 1.7 MB/s)
492
493 -----
494 T E S T S
495 -----
496 Running com.furazin.projecttestgithub.ArithmeticTest
497 Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.091 sec
498
499 Results :
500
501 Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
502
```

Si alguno de los tests fallan, el workflow nos aparecería con error y tendríamos que arreglar el test y volverlo a lanzar para que nos aparezcan todos los jobs completados con éxito.