

API Spring Boot con GitHub Actions y AWS Elastic Beanstalk

Solución de la Actividad 2 - Explicación Técnica del Flujo CI/CD

1. Funcionamiento del Pipeline

El proyecto utiliza GitHub Actions como motor de automatización, definido mediante archivos YAML en .github/workflows/. Se han implementado dos flujos diferenciados:

- **CI (Integración Continua):** Se dispara con cada push o pull_request. Se encarga de clonar el código en un contenedor Ubuntu, configurar el JDK 17 de Amazon Corretto y ejecutar los tests unitarios mediante el Maven Wrapper (./mvnw test). Su objetivo es garantizar que ningún código con errores de compilación o lógica sea aceptado.
- **CD (Despliegue Continuo):** Una vez que el código es válido, este pipeline empaqueta la aplicación en un archivo JAR ejecutable, genera un paquete de despliegue comprimido (ZIP) junto con un archivo Procfile y lo envía a AWS Elastic Beanstalk utilizando las credenciales de AWS Academy almacenadas de forma segura en los GitHub Secrets.

2. Descripción del Flujo CI/CD

El flujo sigue una arquitectura de "entrega garantizada":

1. Desarrollo Local:

El código se escribe y valida en IntelliJ. Se utiliza el Maven Wrapper para asegurar la paridad de versiones de Maven entre el local y la nube.

2. (Trigger):

Al realizar un git push origin main, GitHub detecta los cambios y activa los workflows.

3. Construcción y Artefacto:

El servidor de GitHub genera el archivo binario (.jar) y lo renombra a app.jar para que coincida con la instrucción del Procfile.

4. Despliegue:

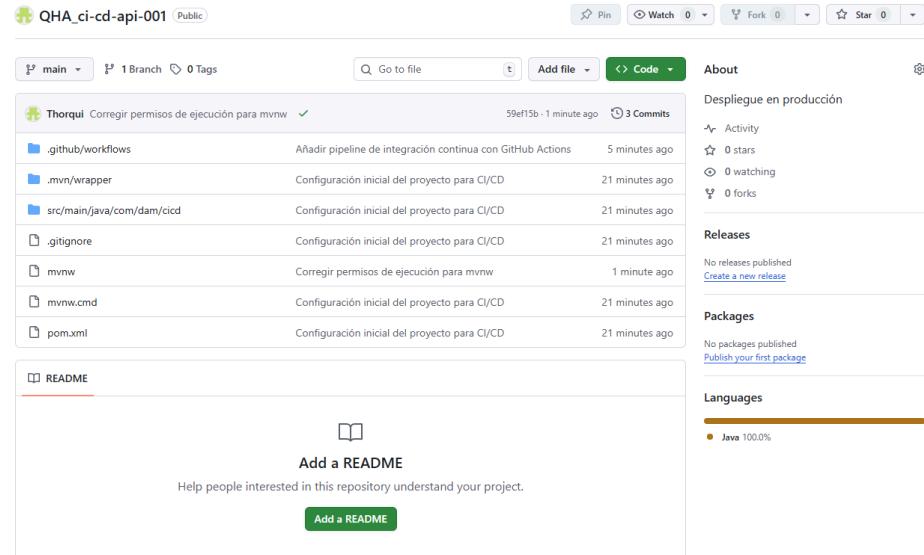
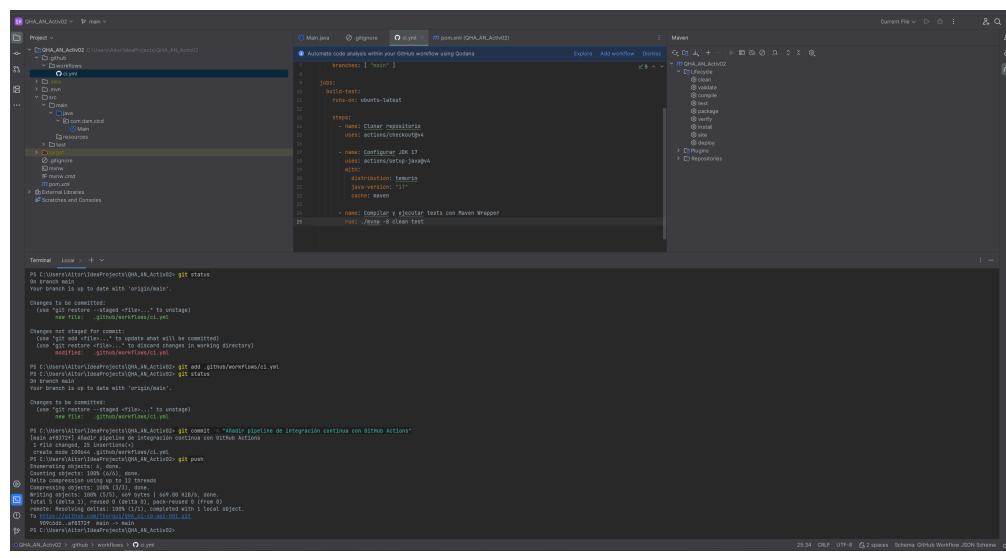
Se utiliza el servicio de AWS Elastic Beanstalk (plataforma Java SE) que recibe el artefacto y actualiza la instancia EC2 sin intervención manual, exponiendo la API en una URL pública. e informe final Excel

3. Decisiones Tomadas y Resolución de Problemas

- **Adaptación a Windows:** Al desarrollar en Windows y desplegar en Linux (AWS), se tomó la decisión de utilizar comandos de PowerShell para la creación de artefactos locales (Compress-Archive) y de gestionar los permisos de ejecución de archivos mediante Git (chmod +x mvnw) para evitar el error de permisos 126 en el pipeline.
 - **Seguridad y Credenciales:** Se decidió utilizar GitHub Secrets para manejar el AWS_SESSION_TOKEN, una medida obligatoria al trabajar con entornos de AWS Academy/Learner Lab, asegurando que las claves no queden expuestas en el código fuente.
 - **Compatibilidad de Puerto:** Se configuró server.port=\${PORT:8080} en las propiedades de Spring Boot. Esta decisión técnica permite que la aplicación funcione en el puerto 8080 localmente pero se adapte dinámicamente al puerto que AWS asigne en producción.

4. Checklist de Entrega:

URL GitHub: https://github.com/Thorqui/QHA_ci-cd-api-001.git



Captura del pipeline en estado Success en GitHub Actions

Dam-ci-ci-apl-001-env

Entorno lanzado correctamente.

Dam-ci-ci-apl-001-env [Información](#)

Acciones ▾ [Cargar e implementar](#)

Información general del entorno

Estado: Pending

ID del entorno: e-igkdm7wq2y

Nombre de aplicación: dam-ci-ci-apl-001

Plataforma: Correto 17 running on 64bit Amazon Linux 2023.4.8.1

Ejecución de la versión: V1.apl-001

Estado de la plataforma: Supported

Eventos Estado Registros Monitoreo Alarmas Actualizaciones administradas Etiquetas

Eventos [10] [Información](#)

Filtrar eventos por texto, propiedad o valor

Hora	Tipo	Detalles
13 de enero de 2026, 08:58:23 (UTC+1)	INFO	Successfully launched environment: Dam-ci-ci-apl-001-env
13 de enero de 2026, 08:58:18 (UTC+1)	INFO	Instance deployment completed successfully.
13 de enero de 2026, 08:37:14 (UTC+1)	INFO	Instance deployment used the commands in your 'Procfile' to initiate startup of your application.
13 de enero de 2026, 08:37:01 (UTC+1)	INFO	Added instance [i-03166fcf580cf88f] to your environment.
13 de enero de 2026, 08:36:12 (UTC+1)	INFO	Waiting for EC2 instances to launch. This may take a few minutes.
13 de enero de 2026, 08:36:01 (UTC+1)	INFO	Environment health has transitioned to Pending. Initialization in progress (running for 20 seconds). There are no instances.
13 de enero de 2026, 08:35:54 (UTC+1)	INFO	Created EIP: 100.50.185.8
13 de enero de 2026, 08:35:38 (UTC+1)	INFO	Created security group named: awseb-e-igkdm7wq2y-stack-AWSEBSecurityGroup-CmMBlBFh5On
13 de enero de 2026, 08:35:12 (UTC+1)	INFO	Using elasticbeanstalk-us-east-1-016533982352 as Amazon S3 storage bucket for environment data.
13 de enero de 2026, 08:35:11 (UTC+1)	INFO	createEnvironment is starting.

No es seguro dam-ci-ci-apl-001-env.eba-2p6pyrd.us-east-1.elasticbeanstalk.com/api/estado

URL pública del entorno de Elastic Beanstalk donde se muestre

Capturas totales realizadas del proceso por fases:

Fase 1: Desarrollo en Local y Entorno

Preparación del entorno de construcción reproducible y validación de la API en local.

Creación y estructura inicial del proyecto base

Verificación de versiones: Java 17, Maven y OpenJDK 17

The screenshot displays two side-by-side IDE windows, likely IntelliJ IDEA, showing Java code and its execution results.

Left IDE (Windows Taskbar at the top):

- Project:** GHL_AN_AutoV02
- Code Editor:** Shows the `Main.java` file with the following code:

```
package com.dcm.click.Main;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Main {
    public static void main(String[] args) {
        SpringApplication.run(Main.class, args);
    }
}
```
- Terminal:** Shows the command `java -jar target/*.jar` being run, resulting in the following output:

```
2023-05-12T10:51:20.484+00:00 INFO 19800 --- [main] com.dcm.click.Main : Starting main v1.0-SNAPSHOT using java 17.0.17 with PID 19800 (C:\Users\Aitor\IdeaProjects\GHL_AN_AutoV02\target\classes started by Aitor in C:\Users\Aitor\IdeaProjects\GHL_AN_AutoV02)
```

Right IDE (Windows Taskbar at the bottom):

- Project:** GHL_AN_AutoV02
- Code Editor:** Shows the `Main.java` file with the same code as the left IDE.
- Maven:** Shows the `pom.xml` file with the following configuration:

```
<parent>com.click</parent>
<groupId>com.dcm.click</groupId>
<artifactId>GHL_AN_AutoV02</artifactId>
<version>1.0-SNAPSHOT</version>
```
- Terminal:** Shows the command `mvn clean package` being run, resulting in the following output:

```
2023-05-12T10:51:20.484+00:00 INFO 19800 --- [main] com.dcm.click.Main : Starting main v1.0-SNAPSHOT using java 17.0.17 with PID 19800 (C:\Users\Aitor\IdeaProjects\GHL_AN_AutoV02\target\classes started by Aitor in C:\Users\Aitor\IdeaProjects\GHL_AN_AutoV02)
```

Ejecución y validación de la aplicación Spring Boot en el entorno local

```

    package org;
    import java.util.*;
    public class Main {
        public static void main(String[] args) {
            System.out.println("Hello World");
        }
    }

```

The Maven panel shows the following configuration:

```

<project>
  <parent>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.8.1</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>QHA_AH_Activ02</artifactId>
  <version>1.0.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>QHA_AH_Activ02</name>
  <description>A simple Java application</description>
  <url>http://www.example.com</url>
  <dependencies>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-api</artifactId>
      <version>5.7.0</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <source>11</source>
          <target>11</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>

```

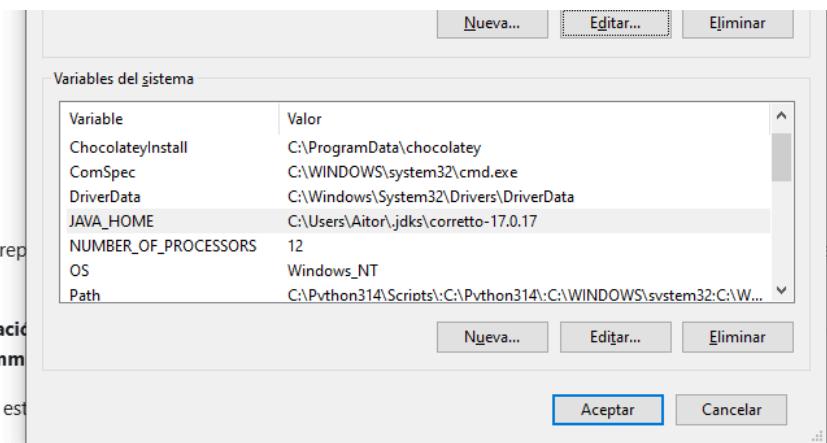
The terminal window shows the build output:

```

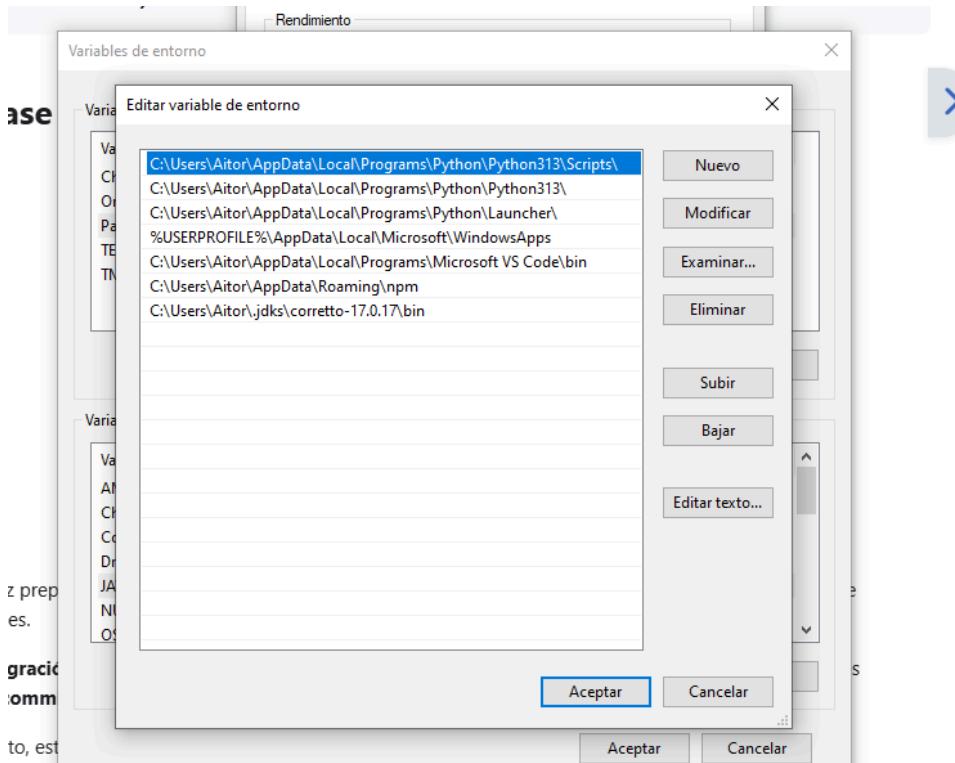
[INFO] Scanning for projects...
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ QHA_AH_Activ02 ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Users\Aitor\IdeaProjects\QHA_AH_Activ02\target\classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] Total time: 0.054 s
[INFO] Final Memory: 14M/200M
[INFO]

```

Resultado satisfactorio del empaquetado (Build Success)



Configuración del repositorio Git



Resolución de problemas en Windows: Configuración de variables de entorno y JAVA_HOME

Fase 2: Control de Versiones (Git)

Inicialización del repositorio y gestión de estados de los archivos.

The screenshot shows the IntelliJ IDEA IDE with the following details:

- Project Tree:** Shows the project structure under "QA_HA_Activ02".
- Main.java:** The current file being edited.
- Terminal:** A terminal window showing the following command-line session:
 - Windows PowerShell
 - Copyright (C) Microsoft Corporation. Todos los derechos reservados.
 - Pruébelo la nueva tecnología PowerShell multipártida <https://aka.ms/azscript>
 - [IntelliJ IDEA] Your PowerShell module version (2.0.0) is outdated, which may cause the plugin with black lines across the terminal screen: https://intellij-support.jetbrains.com/hc/en-us/articles/1130033000000
 - Install the latest version by running: [Install-Module PowerShell -MinimumVersion 2.0.3 -Scope CurrentUser -Force]
 - Install-Module PowerShell -MinimumVersion 2.0.3 -Scope CurrentUser -Force
 - PS C:\Users\alberto\IdeaProjects\QA_HA_Activ02> git status
 - On branch master
 - Untracked files:
 - (use "git add <file>..." to include in what will be committed)
 - .idea/
 - .gitignore
 - .gitmodules
 - pom.xml
 - src/
- Maven:** A tool window showing Maven-related information for the project.

Inicialización del repositorio Git local

The screenshot shows an IDE interface with several panes:

- Project** pane: Shows the project structure for "QHA_AN_Activ02". It includes a "main" package with "Main.java", a ".gitignore" file, and a "pom.xml" file.
- Main.java** code editor: Displays the following Java code:

```
1  # Maven
2  target/
3  !.mvn/wrapper/maven-wrapper.jar
4  !**/src/main/**/target/
5  !**/src/test/**/target/
6
7  # IntelliJ IDEA
8 .idea/
9 *.iml
10 *.im8
11 *.ipr
12
13 # macOS
14 .DS_Store
```
- .gitignore** code editor: Displays the following ignore patterns:

```
*.idea/
*.iml
*.im8
*.ipr
.DS_Store
```
- pom.xml** code editor: Displays the following XML configuration:

```
<!-- https://mvnrepository.com/artifact/com.mycila.maven-dependency-downloader/maven-dependency-downloader -->
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>com.mycila.maven-dependency-downloader</groupId>
            <artifactId>maven-dependency-downloader</artifactId>
            <version>1.0.0</version>
            <scope>runtime</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
```
- Terminal** pane: Shows the command-line history and output of git commands.

```
PS C:\Users\Alitor\IdeaProjects\QHA_AN_Activ02> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    .mvn/
    mvnw
    mvnw.cmd
    pom.xml
    src/

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Alitor\IdeaProjects\QHA_AN_Activ02> git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.mvn/wrapper/MavenWrapperDownloader.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mvnw', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mvnw.cmd', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'pom.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/java/com/mycila/Main.java', LF will be replaced by CRLF the next time Git touches it
PS C:\Users\Alitor\IdeaProjects\QHA_AN_Activ02> git status
On branch master

No commits yet

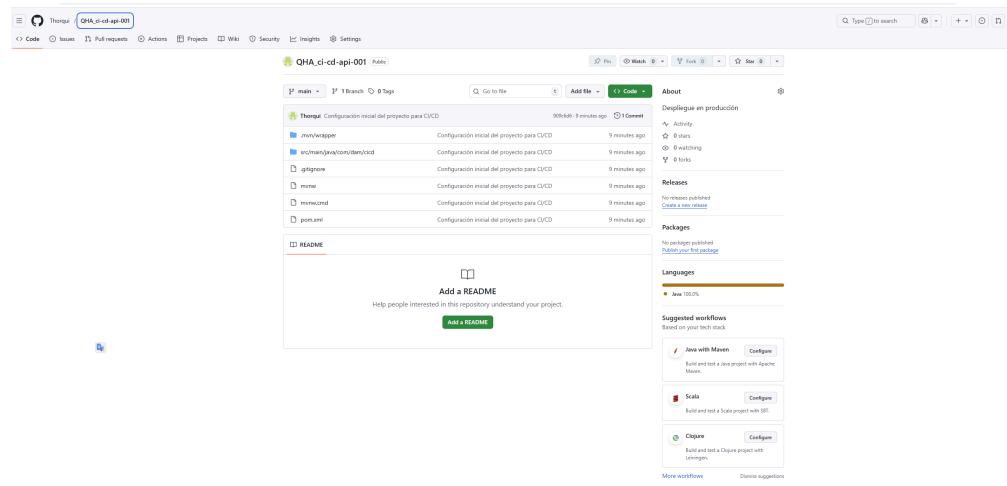
Changes to be committed:
  (use "git rm -cached <file>..." to unstage)
    new file:   .gitignore
    new file:   .mvn/wrapper/maven-dependency-downloader.jar
    new file:   .mvn/wrapper/maven-wrapper.jar
    new file:   .mvn/wrapper/maven-wrapper.properties
    new file:   mvnw
    new file:   mvnw.cmd
    new file:   pom.xml
    new file:   src/main/java/com/ciclo/Main.java
```

Preparación de archivos (Staging) para el primer commit

Seguimiento de la estructura de archivos en el control de versiones

Fase 3: Conexión con GitHub

Configuración del repositorio remoto y sincronización de código



Creación del repositorio vacío en la plataforma GitHub

```
PS C:\Users\Aitor\IdeaProjects\QMA_AN_Activ02> git remote add origin https://github.com/Thorqui/QMA_ci_cd-api-001.git
PS C:\Users\Aitor\IdeaProjects\QMA_AN_Activ02> git branch -M main
PS C:\Users\Aitor\IdeaProjects\QMA_AN_Activ02> git remote -v
origin https://github.com/Thorqui/QMA_ci_cd-api-001.git (fetch)
origin https://github.com/Thorqui/QMA_ci_cd-api-001.git (push)
PS C:\Users\Aitor\IdeaProjects\QMA_AN_Activ02> git push -u origin main
Enumerating objects: 1, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (18/18), 51.85 KiB | 7.41 MiB/s, done.
Total 18 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Thorqui/QMA_ci_cd-api-001.git
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\Aitor\IdeaProjects\QMA_AN_Activ02>
```

Vinculación del repositorio local con el origen remoto

Fase 4: Integración Continua (CI) y Correcciones

Automatización de pruebas y resolución de incompatibilidades Windows-Linux.

QHA_ci_cd-api-001 Public

main 1 Branch 0 Tags

Thorqui Corregir permisos de ejecución para mvnw ✓ 59ef15b · 1 minute ago 3 Commits

.github/workflows Añadir pipeline de integración continua con GitHub Actions 5 minutes ago

.mvn/wrapper Configuración inicial del proyecto para CI/CD 21 minutes ago

src/main/java/com/dam/cicd Configuración inicial del proyecto para CI/CD 21 minutes ago

.gitignore Configuración inicial del proyecto para CI/CD 21 minutes ago

mvnw Corregir permisos de ejecución para mvnw 1 minute ago

mvnw.cmd Configuración inicial del proyecto para CI/CD 21 minutes ago

pom.xml Configuración inicial del proyecto para CI/CD 21 minutes ago

README

Add a README

Help people interested in this repository understand your project.

Add a README

About

Despliegue en producción

Activity

0 stars 0 watching 0 forks

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

Languages

Java 100.0%

Subida de código mediante Personal Access Token

```
PS C:\Users\Aitor\IdeaProjects\QHA_AN_Activ02> git update-index --chmod=+x mvnw
PS C:\Users\Aitor\IdeaProjects\QHA_AN_Activ02> git commit -m "Corregir permisos de ejecución para mvnw"
[main 59ef15b] Corregir permisos de ejecución para mvnw
 1 file changed, 0 insertions(+), 0 deletions(-)
 mode change 100644 => 100755 mvnw
PS C:\Users\Aitor\IdeaProjects\QHA_AN_Activ02> git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 260 bytes | 260.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Thorqui/QHA_ci_cd-api-001.git
 af8372f..59ef15b main -> main
PS C:\Users\Aitor\IdeaProjects\QHA_AN_Activ02>
```

Corrección de permisos de ejecución para mvnw mediante Git update-index

All workflows

Show runs from all workflows

Filter workflow runs

2 workflow runs

	Event	Status	Branch	Actor
Corregir permisos de ejecución para mvnw CI - Construcción con Maven #2; Commit 59ef15b pushed by Thorqui	main	Success	1 minute ago	...
Añadi pipeline de integración continua con GitHub Actions CI - Construcción con Maven #1; Commit af8372f pushed by Thorqui	main	Failure	4 minutes ago	...

Ajuste de flujos de trabajo tras detectar fallos en el pipeline

The screenshot shows the GitHub Actions pipeline editor interface. At the top, there's a header with tabs for 'Main.java', 'gherkin', 'yaml', and 'git commit (GH_AK_Activ2)'. The main workspace contains a single job configuration:

```
stages:
  - name: 'Main'
    steps:
      - name: 'Setup Java'
        uses: actions/setup-java@v3
        with:
          java-version: '17'
      - name: 'Configure JUnit 5'
        uses: actions/configure-junit5@v1
      - name: 'Run tests'
        run: ./mvnw test
      - name: 'Compile y ejecutar tests con Maven Wrapper'
        run: ./mvnw -B clean test
```

On the right side, there are sections for 'Actions' (with a dropdown for 'Main'), 'Logs', 'Artifacts', and 'Metrics'. Below the logs, there's a terminal window showing the command 'git status' and its output. The bottom navigation bar includes links for 'GH_AK_Activ2 > GitHub > workflow > Main'.

Fase 5: Despliegue Continuo (CD) y AWS

Puesta en producción de la API en AWS Elastic Beanstalk.

```

PS C:\Users\Aitor\IdeaProjects\QHA_AN_Activ02> mkdir -p deploy

Directorio: C:\Users\Aitor\IdeaProjects\QHA_AN_Activ02

Mode LastWriteTime Length Name
---- ----- ------
d----- 12/01/2026 10:54 deploy

PS C:\Users\Aitor\IdeaProjects\QHA_AN_Activ02> Copy-Item target/*.jar deploy/app.jar
PS C:\Users\Aitor\IdeaProjects\QHA_AN_Activ02> Copy-Item Procfile deploy/
PS C:\Users\Aitor\IdeaProjects\QHA_AN_Activ02> Compress-Archive -Path deploy/* -DestinationPath deploy.zip -Force
PS C:\Users\Aitor\IdeaProjects\QHA_AN_Activ02> ls

Directorio: C:\Users\Aitor\IdeaProjects\QHA_AN_Activ02

Mode LastWriteTime Length Name
---- ----- ------
d----- 12/01/2026 10:27 github
d----- 12/01/2026 10:33 .mvn
d----- 12/01/2026 9:27 deploy
d----- 12/01/2026 10:58 main
d----- 12/01/2026 9:23 src
d----- 12/01/2026 10:46 .gitignore
d----- 12/01/2026 10:57 155 .gitignore
-a---- 12/01/2026 10:58 18073681 deploy.zip
-a---- 12/01/2026 9:27 10869 avmv
-a---- 12/01/2026 9:27 6697 avmv.cmd
-a---- 12/01/2026 10:40 1455 pom.xml
-a---- 12/01/2026 10:53 22 Procfile

PS C:\Users\Aitor\IdeaProjects\QHA_AN_Activ02>

```

```

# Automate code analysis within your GitHub workflow using Qodana
jobs:
  deploy:
    steps:
      - name: Checkout the repository
        uses: actions/checkout@v4
      # 2. Configurar Java 17
      - name: Configurar Java 17
        uses: actions/setup-java@v4
        with:
          distribution: 'corretto'
          java-version: '17'
          cache: 'maven'
      # 3. Compilar el proyecto y generar el JAR
      - name: Compilar proyecto con Maven
        run: ./mvnw clean package -DskipTests
      # 4. Configurar credenciales de AWS
      - name: Configurar credenciales de AWS
        uses: aws-actions/configure-aws-credentials@v4
        with:
          aws-access-key-id: ${secrets.AWS_ACCESS_KEY_ID}
          aws-secret-access-key: ${secrets.AWS_SECRET_ACCESS_KEY}
          aws-session-token: ${secrets.AWS_SESSION_TOKEN}
          aws-region: ${secrets.AWS_REGION}
      # 5. Preparar el paquete ZIP para Elastic Beanstalk
      - name: Copiar el JAR generado y el Procfile a una carpeta temporal y los comprime
        name: 'Preparar artefacto para Elastic Beanstalk'
        run:
          cp target/*.jar deploy/app.jar
          cp Procfile deploy/Procfile
          cd deploy
          zip -r ..../deploy.zip .
      # 6. Subir nueva versión y desplegar en el entorno de AWS
      - name: Desplegar en Elastic Beanstalk
        uses: timeresponse/com/beanstalk-deploy@v2
        with:
          aws_access_key: ${secrets.AWS_ACCESS_KEY_ID}
          aws_secret_key: ${secrets.AWS_SECRET_ACCESS_KEY}
          session_token: ${secrets.AWS_SESSION_TOKEN}
          region: ${secrets.AWS_REGION}
          environment_name: 'bam-cl-cd-001-env'
          version_label: "v${github.run_id}"
          deployment_package: deploy.zip
          wait_for_environment_recovery: true

```

Preparación de la carpeta de despliegue y configuración del Procfile

Entorno de AWS Elastic Beanstalk configurado y operativo

Verificación de la API accesible públicamente mediante URL de AWS

The screenshot shows the GitHub repository page for QHA_ci-ci-api-001. It displays deployment logs for a pipeline named 'Thompson' with a commit message 'Prueba de despliegue automático: actualización de mensaje'. The deployment status is 'Despliegue en producción'. The repository contains several files including .gitignore, .mvnignore, pom.xml, and various configuration and build scripts.

The screenshot shows the AWS Elastic Beanstalk environment dashboard for 'Dam-ci-ci-api-001-env'. It displays deployment history with 32 events, all marked as successful. The environment ID is e-spkdm/nw2y and the application version is v-20949601685. The platform is Corretto 17 running on 64bit Amazon Linux 2023/4.8.1.

The screenshot shows a terminal window on a Windows machine with the path 'C:\Users\lilien\OneDrive\Proyectos\QHA_AK_Active2'. A 'git push' command is being run, showing the commit message 'Prueba de despliegue automático: actualización de mensaje'. The output shows the commit was successful and pushed to the 'main' branch.

Demostración del ciclo CD: Actualización automática del servicio tras un push en el repositorio