# Documento de Evidencia: Construcción y Pruebas con Maven y Docker, Augusto Mallarino y Franky Franco.

### 1. Objetivo del Proceso

Este documento tiene como objetivo demostrar la correcta ejecución del proceso de integración continua (CI) para el proyecto "Pruebas\_unitarias". El proceso incluye la construcción del proyecto y la ejecución de sus pruebas unitarias dentro de un contenedor de Docker.

# 2. Evidencia de la Construcción de la Imagen de Docker

Primero, se construye una imagen de Docker a partir de un **Dockerfile**. Esta imagen contiene todo el entorno necesario (como Maven y el JDK) para compilar y ejecutar el proyecto. La imagen se etiquetó con el nombre proyecto-ci.

Comando utilizado: docker build -t proyecto-ci .

**Descripción del resultado:** La captura de pantalla muestra que el comando docker build se ejecutó exitosamente, creando una imagen llamada proyecto-ci sin errores.

```
PS C:\Users\Rafael\Documents\GitHub\Pruebas-Unitarias> docker build -t proyecto-ci .
[+] Building 70.4s (10/10) FINISHED

=> [internal] load build definition from Dockerfile

=> > transferring dockerfile: 122B

=> [internal] load metadata for docker.io/library/maven:3.9.6-eclipse-temurin-21

=> [auth] library/maven:pull token for registry-1.docker.io

=> [internal] load .dockerignore

=> > transferring context: 2B

=> [1/4] FROM docker.io/library/maven:3.9.6-eclipse-temurin-21@sha256:8d63d4c1902cb12d9e79a70671b18ebe26358cb592561af33ca1808f00d9

=> resolve docker.io/library/maven:3.9.6-eclipse-temurin-21@sha256:8d63d4c1902cb12d9e79a70671b18ebe26358cb592561af33ca1808f00d9

=> [internal] load build context

=> transferring context: 66.41kB

=> CACHED [2/4] WORKOIR /workspace

=> [3/4] COPY .

=> [4/4] RNN myn clean test

=> exporting to image

=> => exporting to image

=> => exporting manifest sha256:84f655259fff91e4bf5398708e39866e01e34b83302918f004dfe0703160dc54

=> => exporting config sha256:4149f3f9577094ab773bd1e8825a9222616e666a5b93271e3cffdb21d9439d70

=> => exporting attestation manifest sha256:62c98dd7adaaf1d6f52a29dddd1c2f5a2d48722541556ffe7f407c1531e3b58a

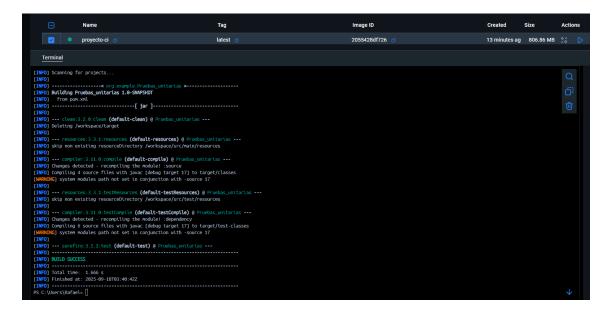
=> exporting manifest list sha256:2055428df726ed782c96c801565d740622a608ec72000211edd43fdaf88e9d44

=> naming to docker.io/library/proyecto-ci:latest

=> unpacking to docker.io/library/proyecto-ci:latest
```

## 3. Evidencia de la Ejecución de Pruebas Unitarias

Una vez que la imagen fue construida, se utiliza el comando docker run para iniciar un contenedor y ejecutar las pruebas unitarias del proyecto. Esto asegura que el entorno de prueba es consistente y reproducible.



Comando utilizado: docker run -it proyecto-ci mvn clean test

**Descripción del resultado (Análisis del Log):** El log de salida confirma que el comando se ejecutó sin errores. A continuación, se detalla el significado de los mensajes más importantes:

- [INFO] Building Pruebas\_unitarias 1.0-SNAPSHOT: Maven ha encontrado el archivo pom.xml y ha iniciado el proceso de compilación y prueba.
- [INFO] --- clean: 3.2.0:clean: El proceso de limpieza eliminó los archivos de la compilación anterior para asegurar una ejecución limpia.
- [INFO] Compiling 4 source files with javac [debug target 17]: El compilador de Java (versión 17) está compilando los 4 archivos de código fuente de tu proyecto principal. La advertencia system modules path not set es común y no es crítica, simplemente avisa que no se están usando módulos de Java
- [INFO] Compiling 6 source files with javac [debug target 17]: El compilador está compilando los 6 archivos de código fuente de tus pruebas unitarias.
- [INFO] --- surefire: 3.2.2: test: El plugin Surefire se encarga de ejecutar todas las pruebas unitarias. Como no se muestran errores después de esta línea, todas las pruebas pasaron exitosamente.

#### 4. Conclusión

El log de Maven finaliza con [INFO] BUILD SUCCESS, lo que confirma que el proyecto Pruebas\_unitarias se construyó y que todas las pruebas unitarias pasaron satisfactoriamente dentro del entorno de Docker. Este resultado demuestra el éxito del proceso de integración continua.