

Asignatura	Datos del alumno	Fecha
<b>Entornos de Integración y Entrega Continua</b>	<b>Apellidos:</b> Brito Casanova	16/06/2024
	<b>Nombres:</b> Geovanny José	

### Laboratorio: Jenkinsfile

Usar el Jenkinsfile del primer caso práctico como punto de partida y añadir lo siguiente:

- ▶ Nuevas etapas para ejecutar las pruebas de API y las pruebas de E2E.
- ▶ Archivado de los archivos XML de pruebas.
- ▶ Presentación de los informes de pruebas de API y E2E.
- ▶ Al finalizar, enviar un correo indicando que el trabajo ha terminado solo si el *pipeline* ha resultado en fallo. El correo debe incluir el nombre del trabajo y el número de ejecución.

### Rúbrica

Laboratorio: Jenkinsfile	Descripción	Puntuación máxima	Peso
Criterio 1	El Jenkinsfile incluye todas las fases necesarias.	4	40%
Criterio 2	La fase post contiene el envío de correo correctamente.	4	40%
Criterio 3	El Jenkinsfile se ejecuta correctamente.	2	20%
		<b>10</b>	<b>100</b>

Asignatura	Datos del alumno	Fecha
<b>Entornos de Integración y Entrega Continua</b>	<b>Apellidos:</b> Brito Casanova	16/06/2024
	<b>Nombres:</b> Geovanny José	

## Tabla de Contenido

Laboratorio: Jenkinsfile	1
Desarrollo de la actividad	3
1. Fork del repositorio	3
1.1. Estructura del proyecto modificado	3
2. Creación de Vagrant	3
2.1. Código Vagrant	3
2.2. Resultado de la ejecución del Vagrantfile	5
3. Configuración de Jenkins	5
3.1. Instalación de plugins sugeridos	5
3.2. Página principal de Jenkins	6
3.3. Instalación de plugins adicionales	6
4. Fases del Jenkins	6
4.1. Creación de nueva tarea tipo Pipeline	6
4.2. Pipeline Script	6
4.3. Configuración del Pipeline	7
4.4. Listado de las tareas	8
4.5. Construcción de la tarea	8
4.6. Resultado de la tarea	8
4.6.1. Resultado: Pipeline Console	9
4.6.2. Resultado: Console Output	9
4.6.3. Resultado: Pipeline Overview	9
4.6.4. Resultado: Pipeline Steps	9
Conclusiones	10
Anexos	11

Asignatura	Datos del alumno	Fecha
Entornos de Integración y Entrega Continua	Apellidos: Brito Casanova	16/06/2024
	Nombres: Geovanny José	

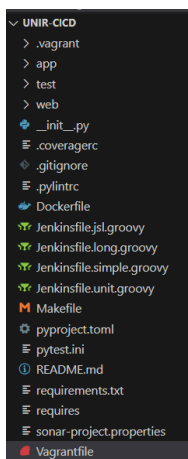
## Desarrollo de la actividad

### 1. Fork del repositorio

La documentación y todos los cambios que se realizaron para el cumplimiento de los criterios de la actividad, se encuentra en el siguiente enlace:

<https://github.com/geo-bricex/unir-cicd.git>

#### 1.1. Estructura del proyecto modificado



### 2. Creación de Vagrant

Para facilidad de la actividad se creó un Vagrantfile, que automatice la creación de una máquina virtual con todo lo necesario para completar la actividad.

#### 2.1. Código Vagrant

```

1. # Configuración de Vagrant para instalar Jenkins en una máquina virtual con
   Ubuntu 20.04
2. Vagrant.configure("2") do |config|
3.   # Utilizar la caja base de Ubuntu 20.04
4.   config.vm.box = "ubuntu/focal64"
5.
6.   # Configurar la red para permitir el acceso a Jenkins desde el host
7.   # Redirigir el puerto 8080 en la VM al puerto 8089 en el host
8.   config.vm.network "forwarded_port", guest: 8080, host: 8089
9.   config.ssh.insert_key = false
10.
11.  # Aumentar los recursos asignados a la máquina virtual
12.  config.vm.provider "virtualbox" do |vb|
13.    vb.memory = "8192" # Aumentar la memoria RAM a 8192 MB (8 GB)
14.    vb.cpus = 4         # Aumentar el número de CPUs a 4
15.  end
16.
17.  # Provisión para configurar SSH, Docker, Jenkins y make utilizando un script
   de shell
18.  config.vm.provision "shell", inline: <<-SHELL
19.    # Actualizar la lista de paquetes
20.    sudo apt-get update
21.

```

Asignatura	Datos del alumno	Fecha
<b>Entornos de Integración y Entrega Continua</b>	<b>Apellidos:</b> Brito Casanova	16/06/2024
	<b>Nombres:</b> Geovanny José	

```

22. # Instalar Java, que es un requisito necesario para Jenkins
23. sudo apt-get install -y openjdk-11-jdk
24.
25. # Instalar Docker
26. sudo apt-get install -y apt-transport-https ca-certificates curl software-
properties-common
27. curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add
-
28. sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
29. sudo apt-get update
30. sudo apt-get install -y docker-ce
31.
32. # Agregar el usuario vagrant y jenkins al grupo docker
33. sudo usermod -aG docker vagrant
34. sudo usermod -aG docker jenkins
35.
36. # Instalar make y otras herramientas necesarias
37. sudo apt-get install -y build-essential
38.
39. # Descargar la clave GPG de Jenkins y añadirla al keyring del sistema
40. sudo wget -O /usr/share/keyrings/jenkins-keyring.asc
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
41.
42. # Añadir el repositorio de Jenkins a la lista de fuentes de APT
43. echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]
https://pkg.jenkins.io/debian-stable binary/" |
44. sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
45.
46. # Actualizar la lista de paquetes nuevamente para incluir Jenkins
47. sudo apt-get update
48.
49. # Instalar Jenkins
50. sudo apt-get install -y jenkins
51.
52. # Iniciar el servicio de Jenkins
53. sudo systemctl start jenkins
54.
55. # Habilitar Jenkins para que se inicie automáticamente al arrancar el
sistema
56. sudo systemctl enable jenkins
57.
58. # Configurar SSH para permitir la autenticación por contraseña
59. sudo sed -i 's/PasswordAuthentication no/PasswordAuthentication yes/'
/etc/ssh/sshd_config
60. sudo sed -i 's/ChallengeResponseAuthentication
yes/ChallengeResponseAuthentication no/' /etc/ssh/sshd_config
61. echo "AllowUsers vagrant jenkins" | sudo tee -a /etc/ssh/sshd_config
62. echo "UsePAM yes" | sudo tee -a /etc/ssh/sshd_config
63. sudo systemctl restart ssh
64.
65. # Establecer la contraseña para el usuario vagrant
66. echo "vagrant:vagrant" | sudo chpasswd
67.
68. # Reiniciar el servicio de Jenkins para aplicar los cambios
69. sudo systemctl restart jenkins
70.
71. sudo apt update
72. sudo apt install python3-pip -y
73. pip3 install flask

```

Asignatura	Datos del alumno	Fecha
Entornos de Integración y Entrega Continua	Apellidos: Brito Casanova	16/06/2024
	Nombres: Geovanny José	

```

74.
75. SHELL
76.
77. # Configuración opcional para sincronizar una carpeta del host con la VM
78. config.vm.synced_folder ".", "/vagrant"
79. end

```

## 2.2. Resultado de la ejecución del Vagrantfile

```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
default: Setting up jenkins (2.452.2) ...
default: Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
default: Processing triggers for man-db (2.9.1-1) ...
default: Processing triggers for systemd (245.4-4ubuntu3.23) ...
default: Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
default: Executing: /lib/systemd/systemd-sysv-install enable jenkins
default: AllowUsers vagrant
default: UsePAM yes
PS C:\Users\geova\OneDrive\Estudios Superiores\UNIR - POSGRADO\Trabajo Fin de Master\Scripts\vagrant\ubuntu\jenkins>

```

## 3. Configuración de Jenkins

Accedemos al enlace que se configuró para Jenkins desde nuestra máquina host

<http://localhost:8089>

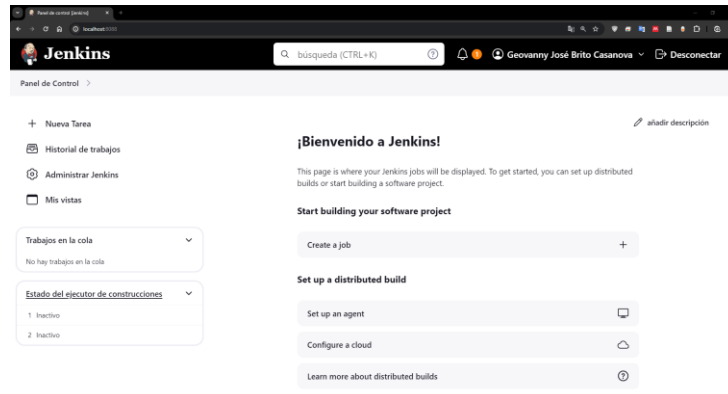
### 3.1. Instalación de plugins sugeridos



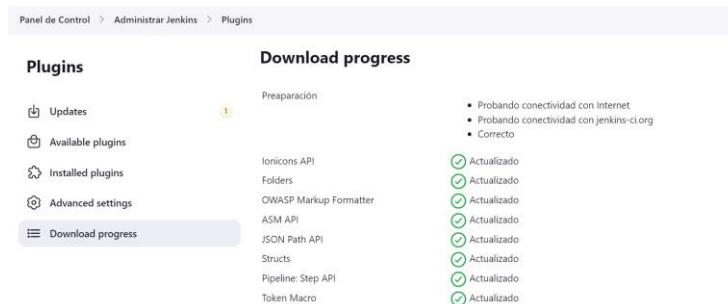
Posteriormente, llenamos información requerida, como la contraseña administradora inicial, proporcionamos nuevas credenciales para el inicio de sesión, así como más información personal, etc.

Asignatura	Datos del alumno	Fecha
Entornos de Integración y Entrega Continua	Apellidos: Brito Casanova	16/06/2024
	Nombres: Geovanny José	

### 3.2. Página principal de Jenkins

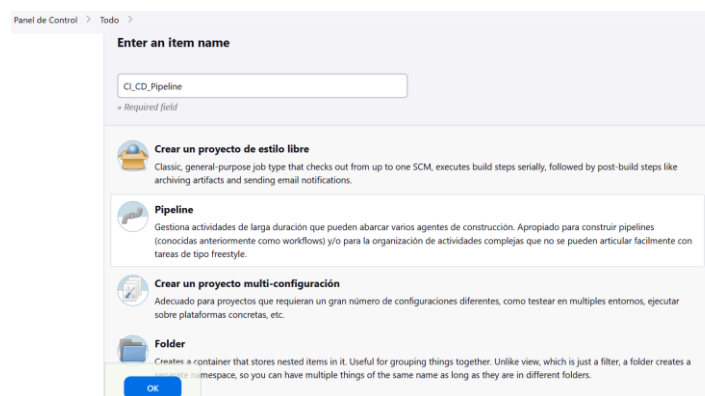


### 3.3. Instalación de plugins adicionales



## 4. Fases del Jenkins

### 4.1. Creación de nueva tarea tipo Pipeline



### 4.2. Pipeline Script

```

1. pipeline {
2.   agent any
3.   stages {
4.     stage('Checkout') {
5.       steps {
6.         git 'https://github.com/geo-bricex/unir-cicd.git'
7.       }
8.     }
9.     stage('Build') {

```

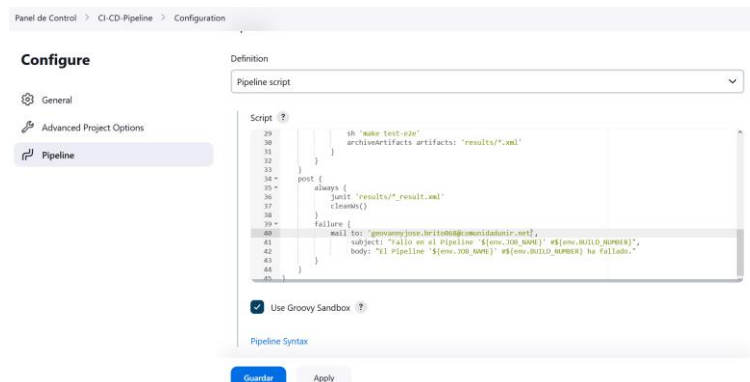
Asignatura	Datos del alumno	Fecha
Entornos de Integración y Entrega Continua	Apellidos: Brito Casanova	16/06/2024
	Nombres: Geovanny José	

```

10.     steps {
11.         echo 'Building stage!'
12.         sh 'make build'
13.     }
14. }
15. stage('Unit Tests') {
16.     steps {
17.         sh 'make test-unit'
18.         archiveArtifacts artifacts: 'results/*.xml'
19.     }
20. }
21. stage('API Tests') {
22.     steps {
23.         sh 'make test-api'
24.         archiveArtifacts artifacts: 'results/*.xml'
25.     }
26. }
27. stage('E2E Tests') {
28.     steps {
29.         sh 'make test-e2e'
30.         archiveArtifacts artifacts: 'results/*.xml'
31.     }
32. }
33. }
34. post {
35.     always {
36.         junit 'results/*_result.xml'
37.         cleanWs()
38.     }
39.     failure {
40.         mail to: 'geovannyjose.brito068@comunidadunir.net',
41.             subject: "Fallo en el Pipeline '${env.JOB_NAME}' #${env.BUILD_NUMBER}",
42.             body: "El Pipeline '${env.JOB_NAME}' #${env.BUILD_NUMBER} ha fallado."
43.     }
44. }
45. }
46.

```

### 4.3. Configuración del Pipeline



Asignatura	Datos del alumno	Fecha
Entornos de Integración y Entrega Continua	Apellidos: Brito Casanova	16/06/2024
	Nombres: Geovanny José	

4.4. Listado de las tareas

Nueva Tarea

Historial de trabajos

Administrar Jenkins

Mis vistas

Trabajos en la cola

No hay trabajos en la cola

Estado del ejecutor de construcciones

built-in node (0 of 2 executors busy)

Panel de Control

busqueda (CTRL+K)

Geovanny

Desconectar

Todo

+

S	W	Nombre	Último Exito	Último Fallo	Última Duración
✓	☀	CI-CD-Pipeline	29 Min #1	N/D	1 Min 23 Seg
✓	☁	FirstPipeline	1 Hor 20 Min #67	1 Hor 28 Min #65	1 Min 37 Seg

Iconos: S M L

REST API Jenkins 2.452.2

4.5. Construcción de la tarea

Build #1 (16 jun. 2024 7:15:35)

Conservar esta ejecución para siempre

Changelog

Console Output

Edit Build Information

Delete build #1

Timings

Git Build Data

Resultado de los tests

Pipeline Overview

Pipeline Console

Restart from Stage

Replay

Pipeline Steps

Workspaces

Build Artifacts

api\_result.xml 306 B view

coverage.xml 2.06 KB view

cypress\_result.xml 947 B view

unit\_result.xml 1.02 KB view

Iniciado por el usuario Geovanny

This run spent:

- 30 Mo waiting
- 1 Min 23 Seg build duration
- 1 Min 24 Seg total from scheduled to completion

Revisión: 5042f3a3e72ae78c715b4b03a06c4dd4b0d698

Repository: https://github.com/geov-briceu/unir-cicd.git

refs/remotes/origin/master

Resultado de los tests (Sin fallos)

4.6. Resultado de la tarea

CI-CD-Pipeline

Desactivar el Proyecto

Changelog

Construir ahora

Configurar

Borrar Pipeline

Stages

Rename

Pipeline Syntax

Last Successful Artifacts

api\_result.xml 306 B view

coverage.xml 2.06 KB view

cypress\_result.xml 947 B view

unit\_result.xml 1.02 KB view

Últimos resultados de tests (Sin fallos)

Enlaces permanentes

• "Última ejecución (#1) hace 31 Min"

• "Última ejecución estable (#1) hace 31 Min"

• "Última ejecución correcta (#1) hace 31 Min"

• "Last completed build (#1) hace 31 Min"

Historia de tareas

Tendencia

Filter...

16 jun. 2024 7:13

Atom feed para todos

Atom feed para los admins

Tendencia de los resultados de pruebas

Passed Skipped Failed

12 13

11 0

10 0

9 0

8 0

7 0

13 0 0

© Universidad Internacional de La Rioja (UNIR)

Actividades

8



Asignatura	Datos del alumno	Fecha
Entornos de Integración y Entrega Continua	Apellidos: Brito Casanova	16/06/2024
	Nombres: Geovanny José	

#### 4.6.1. Resultado: Pipeline Console

Panel de Control > CI-CD-Pipeline > #1 > Pipeline Console

**Build #1** Rebuild Overview Configure ...

Correcto 32 Min ago in 1 Min 23 Seg

- Checkout
- Build
- Unit Tests
- API Tests
- E2E Tests
- Post Actions**

**Stage 'Post Actions'**

- Started 31 Min ago
- Queued 8 Ms
- Took 0.13 Seg
- Success
- [View as plain text](#)

**results"/ result.xml** 59 Ms 🔍 🔗 📄

ArchiveJUnit:formatted test results

0 Recording test results

1 [Checks API] No suitable checks publisher found.

**Delete workspace when build is done** 40 Ms 🔍 🔗 📄

0 [MS-CLEANUP] Deleting project workspace...

1 [MS-CLEANUP] Deferred wipeout is used...

2 [MS-CLEANUP] done

#### 4.6.2. Resultado: Console Output

```
docker cp e2e-tests:/results ./ || true
docker rm --force apiserver || true
apiserver
docker rm --force calc-web || true
calc-web
docker rm --force e2e-tests || true
e2e-tests
docker network rm calc-test-e2e || true
calc-test-e2e
[Pipeline] archiveArtifacts
Archiving artifacts
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] junit
Recording test results
[Checks API] No suitable checks publisher found.
[Pipeline] cleanWs
[MS-CLEANUP] Deleting project workspace...
[MS-CLEANUP] Deferred wipeout is used...
[MS-CLEANUP] done
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

#### 4.6.3. Resultado: Pipeline Overview

Panel de Control > CI-CD-Pipeline > #1 > Pipeline Overview

**Build #1** Rebuild Console Configure

**Pipeline**

Start → Checkout → Build → Unit Tests → API Tests → E2E Tests → Post Actions → End

**Details**

- Manually run by Geovanny
- Started 35 Min ago
- Queued 6 Ms
- Took 1 Min 23 Seg

#### 4.6.4. Resultado: Pipeline Steps

archiveArtifacts - (36 Ms in self)	results"/ result.xml	🔍	📄
stage - (15 Seg in block)	API Tests	🔍	📄
stage block (API Tests) - (15 Seg in block)			
sh - (15 Seg in self)	make test-api	🔍	📄
archiveArtifacts - (34 Ms in self)	results"/ result.xml	🔍	📄
stage - (33 Seg in block)	E2E Tests	🔍	📄
stage block (E2E Tests) - (33 Seg in block)			
sh - (33 Seg in self)	make test-e2e	🔍	📄
archiveArtifacts - (37 Ms in self)	results"/ result.xml	🔍	📄
stage - (0.17 Seg in block)	Declarative: Post Actions	🔍	📄
stage block (Declarative: Post Actions) - (0.13 Seg in block)			
junit - (59 Ms in self)	results"/ result.xml	🔍	📄
cleanWs - (40 Ms in self)		🔍	📄

Asignatura	Datos del alumno	Fecha
<b>Entornos de Integración y Entrega Continua</b>	<b>Apellidos:</b> Brito Casanova	16/06/2024
	<b>Nombres:</b> Geovanny José	

### Conclusiones

Se ha logrado cumplir con el primer criterio de evaluación, ya que el Jenkinsfile modificado incluye todas las etapas requeridas. Se añadieron nuevas fases para la ejecución de pruebas de API y pruebas de extremo a extremo (E2E), siguiendo las directrices establecidas en la actividad. Estas etapas adicionales garantizan que se realicen pruebas completas del sistema, incrementando la fiabilidad del pipeline y asegurando que todos los componentes funcionen correctamente antes de la integración y entrega continua.

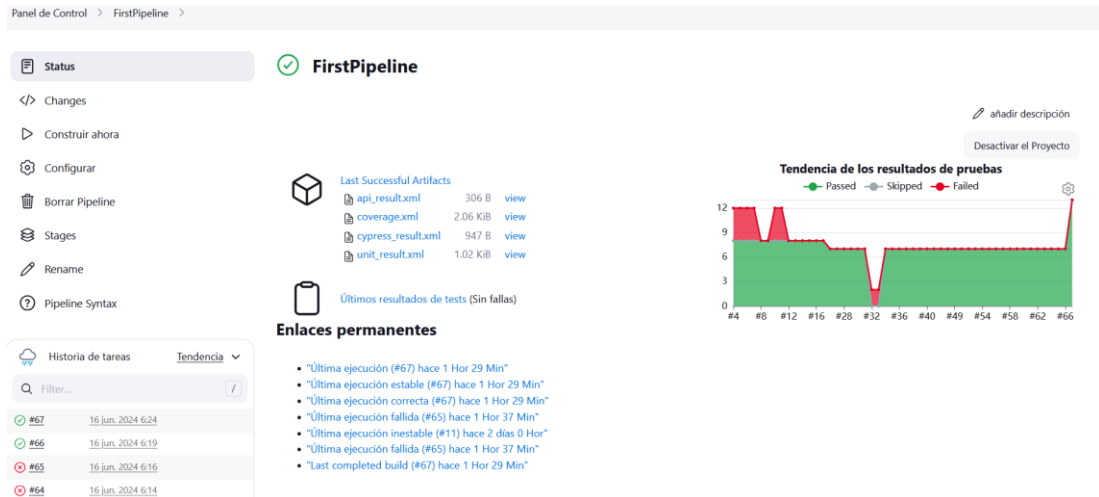
El segundo criterio se cumple al incluir el envío de correos en la fase post. En caso de que el pipeline falle, se ha configurado el envío de un correo electrónico que notifica sobre el fallo, incluyendo el nombre del trabajo y el número de ejecución. Esta funcionalidad es esencial para mantener una comunicación efectiva sobre el estado del pipeline y permite una rápida reacción ante cualquier problema que se presente, facilitando la gestión y el seguimiento de los proyectos.

Finalmente, el tercer criterio también se cumple, ya que el Jenkinsfile se ejecuta correctamente. Las capturas de pantalla adjuntas en el informe muestran el proceso de definición del trabajo en Jenkins, así como la vista del Stage View con una ejecución satisfactoria. La correcta ejecución del pipeline confirma que todas las configuraciones y etapas se han implementado adecuadamente, y que el sistema funciona según lo esperado, permitiendo la integración y entrega continua de manera eficiente.

Asignatura	Datos del alumno	Fecha
Entornos de Integración y Entrega Continua	Apellidos: Brito Casanova	16/06/2024
	Nombres: Geovanny José	

## Anexos

- Para obtener los resultados esperados de la actividad se necesitó de 67 construcciones.



- Otro PipelineScript con más validaciones:

```

1. pipeline {
2.   agent any
3.   stages {
4.     stage('Checkout') {
5.       steps {
6.         git 'https://github.com/geo-bricex/unir-cicd.git'
7.       }
8.     }
9.     stage('Build') {
10.      steps {
11.        echo 'Building stage!'
12.        sh 'make build'
13.      }
14.    }
15.    stage('Unit Tests') {
16.      steps {
17.        sh 'make test-unit'
18.        archiveArtifacts artifacts: 'results/unit_result.xml'
19.      }
20.    }
21.    stage('API Tests') {
22.      steps {
23.        script {
24.          sh 'docker network create calc-test-api || true'
25.          sh 'docker run --network calc-test-api --name apiserver --entrypoint=calculator-app:latest mkdir -p /opt/calc/results'
26.          sh 'docker run -d --network calc-test-api --env PYTHONPATH=/opt/calc --name apiserver --env FLASK_APP=app/api.py -p 5000:5000 calculator-app:latest flask run --host=0.0.0.0 || true'
27.          sh 'docker run --network calc-test-api --name api-tests --env PYTHONPATH=/opt/calc --env BASE_URL=http://apiserver:5000/ calculator-app:latest pytest --junit-xml=results/api_result.xml -m api || true'
28.          sh 'docker cp api-tests:/opt/calc/results . || true'
29.          sh 'docker stop apiserver || true'
30.          sh 'docker rm --force apiserver || true'
31.          sh 'docker stop api-tests || true'

```

Asignatura	Datos del alumno	Fecha
<b>Entornos de Integración y Entrega Continua</b>	<b>Apellidos:</b> Brito Casanova	16/06/2024
	<b>Nombres:</b> Geovanny José	

```

32.         sh 'docker rm --force api-tests || true'
33.         sh 'docker network rm calc-test-api || true'
34.     }
35.     archiveArtifacts artifacts: 'results/api_result.xml'
36. }
37. }
38. stage('E2E Tests') {
39.     steps {
40.         script {
41.             sh 'docker network create calc-test-e2e || true'
42.             sh 'docker stop apiserver || true'
43.             sh 'docker rm --force apiserver || true'
44.             sh 'docker stop calc-web || true'
45.             sh 'docker rm --force calc-web || true'
46.             sh 'docker stop e2e-tests || true'
47.             sh 'docker rm --force e2e-tests || true'
48.             sh 'docker run -d --network calc-test-e2e --env PYTHONPATH=/opt/calc
--name apiserver --env FLASK_APP=app/api.py -p 5000:5000 calculator-app:latest flask run --
host=0.0.0.0'
49.             sh 'docker run -d --network calc-test-e2e --name calc-web -p 80:80
calc-web'
50.             sh 'docker create --network calc-test-e2e --name e2e-tests
cypress/included:4.9.0 --browser chrome || true'
51.             sh 'docker cp ./test/e2e/cypress.json e2e-tests:/cypress.json'
52.             sh 'docker cp ./test/e2e/cypress e2e-tests:/cypress'
53.             sh 'docker start -a e2e-tests || true'
54.             sh 'docker cp e2e-tests:/results ./ || true'
55.             sh 'docker rm --force apiserver || true'
56.             sh 'docker rm --force calc-web || true'
57.             sh 'docker rm --force e2e-tests || true'
58.             sh 'docker network rm calc-test-e2e || true'
59.         }
60.         archiveArtifacts artifacts: 'results/e2e_result.xml'
61.     }
62. }
63. }
64. post {
65.     always {
66.         junit 'results/*_result.xml'
67.         cleanWs()
68.     }
69.     failure {
70.         mail to: 'your@email.com',
71.             subject: "Fallo en el Pipeline '${env.JOB_NAME}' #${env.BUILD_NUMBER}",
72.             body: "El Pipeline '${env.JOB_NAME}' #${env.BUILD_NUMBER} ha fallado."
73.     }
74.     success {
75.         mail to: 'your@email.com',
76.             subject: "Éxito en el Pipeline '${env.JOB_NAME}' #${env.BUILD_NUMBER}",
77.             body: "El Pipeline '${env.JOB_NAME}' #${env.BUILD_NUMBER} se ha
ejecutado con éxito."
78.     }
79. }
80. }

```

El pipeline ha cumplido con todas las etapas de la actividad (checkout, build, unit tests, API tests, y E2E tests) sin presentar errores debido a la correcta implementación de cada etapa y la gestión adecuada de los recursos Docker. Esto asegura que todas las pruebas se hayan ejecutado correctamente y los resultados se hayan archivado, logrando así un pipeline de CI/CD robusto y funcional.