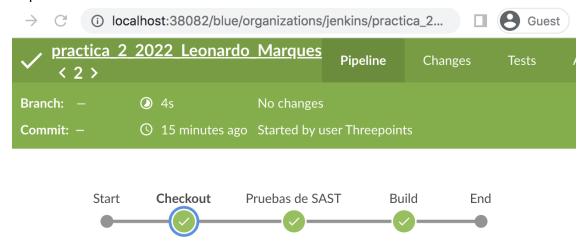
01 - Creación de un Pipeline Declarativo básicode Cl

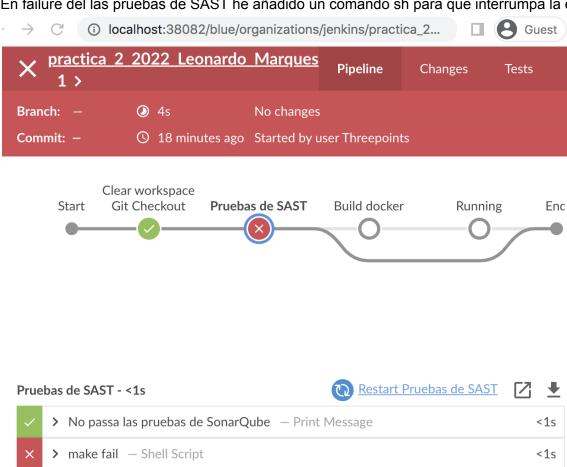
a. Codificación del pipeline:

```
pipeline {
    agent any
    environment {
        PASS_QUALITY_GATE = 'False'
    }
    stages {
        stage('Checkout') {
            steps {
                cleanWs()
                echo 'Checking out'
                git credentialsId: 'e9b5c362-f040-4029-b41f-7e0b64015e87',
url: 'git@localhost:threepoints/docker-vulnerable-dvwa.git'
        stage('Pruebas de SAST'){
            steps {
                script {
                    if (PASS_QUALITY_GATE == 'True') {
                        echo 'Pasa las pruebas de SonarQube'
                    } else {
                        echo 'No pasa las pruebas de SonarQube'
                        sh 'make fail'
                    }
                }
            }
        }
        stage('Build'){
            steps{
                sh 'docker build -t dvwa .'
            }
       }
   }
}
```

b. Capturas OceanBlue



En failure del las pruebas de SAST he añadido un comando sh para que interrumpa la ejecución



c. Capturas SonarQube "Mocked"

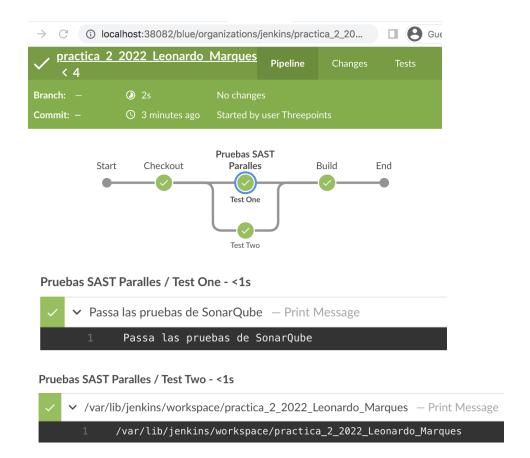
```
[Pipeline] stage
                                     [PIPeIIne] Stage
                                     [Pipeline] { (Pruebas de SAST)
[Pipeline] { (Pruebas de SAST)
                                     [Pipeline] script
[Pipeline] script
                                     [Pipeline] {
[Pipeline] {
                                     [Pipeline] echo
[Pipeline] echo
                                     No passa las pruebas de SonarQube
Passa las pruebas de SonarQube
                                     [Pipeline] sh
[Pipeline] }
                                     + make fail
[Pipeline] // script
                                     /var/lib/jenkins/workspace/practica_2_2
[Pipeline] }
                                     [Pipeline] }
[Pipeline] // stage
                                     [Pipeline] // script
```

02. Ejecutar dos steps de manera paralela.

a. Codificación del pipeline

```
stage('Pruebas SAST Paralles') {
            parallel {
                stage('Test One') {
                    steps {
                        script {
                             if (PASS_QUALITY_GATE == 'True') {
                                 echo 'Pasa las pruebas de SonarQube'
                             } else {
                                 echo 'No pasa las pruebas de SonarQube'
                                 //sh 'make fail'
                                 currentBuild.result = 'FAILURE'
                        }
                    }
                }
                stage('Test Two') {
                    steps {
                        echo "${WORKSPACE}"
                    }
                }
            }
        }
```

b. Capturas OceanBlue.



c. Capturas de la consola.

```
[Pipeline] stage
           [Pipeline] { (Pruebas SAST Paralles)
           [Pipeline] parallel
           [Pipeline] { (Branch: Test One)
           [Pipeline] { (Branch: Test Two)
           [Pipeline] stage
           [Pipeline] { (Test One)
           [Pipeline] stage
           [Pipeline] { (Test Two)
           [Pipeline] script
           [Pipeline] {
           [Pipeline] echo
[Test Two] /var/lib/jenkins/workspace/practica_2_2022_Leonardo_Marques
           [Pipeline] }
           [Pipeline] echo
[Test One] Passa las pruebas de SonarQube
           [Pipeline] }
           [Pipeline] // stage
```

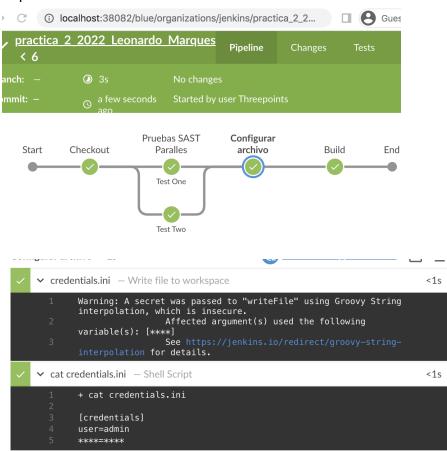
03. Usar credenciales en un pipeline declarativo.

a. Codificación del pipeline:

En el ejercicio creo que no es lo esperado, he tenido errores con la escrita en el fichero y los datos sensibles como la clave.

Credentials_Threepoints admin/*****

b. Capturas OceanBlue:



c. Capturas de la consola:

```
[Pipeline] stage
[Pipeline] { (Configurar archivo)
[Pipeline] withCredentials
Masking supported pattern matches of $password
[Pipeline] {
[Pipeline] writeFile
Warning: A secret was passed to "writeFile" using Groovy String interpolation, which is insecure.
                Affected argument(s) used the following variable(s): [****]
                 See https://jenkins.io/redirect/groovy-string-interpolation for details.
[Pipeline] sh
+ cat credentials.ini
[credentials]
user=admin
****=***
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
```