Propuesta de solución

Caso Práctico 1 – Apartado B

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Leon Granda	
	Nombre: Giovanna Victoria	

URL de repositorio solución de GitHub: https://github.com/GiovannaLeon/helloworld.git

Reto 1 – Creación pipeline CI

En este reto se solicitan 4 entregables:

- URL al repositorio creado por el alumno, a partir del código fuente base de este CP1, que albergue tanto el código fuente como el Jenkinsfile. (jenkinsfile:

```
pipeline {
  agent any

  stages {
    stage('Get Code') {
      steps {
        git 'https://github.com/GiovannaLeon/helloworld.git'
        bat "dir"
        echo WORKSPACE
      }
    }

    // Etapa de Pruebas Unitarias
    stage('Unit') {
      steps {
        catchError(buildResult: 'UNSTABLE', stageResult: 'FAILURE') {
          bat '''
            SET PYTHONPATH=%WORKSPACE%
            C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\python.exe -m pytest --junitxml=result-unit.xml test\\unit
          '''
          sleep(5)
          junit 'result*.xml' // Reporte de las pruebas unitarias
        }
      }
    }

    // Etapa de Cobertura de Pruebas
    stage('Coverage') {
      steps {
        bat '''
          C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\Scripts\\coverage.exe run --branch --source=app --
omit=app\\__init__.py,app\\api.py -m pytest test\\unit
          C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\Scripts\\coverage.exe xml
        '''
        catchError(buildResult: 'UNSTABLE', stageResult: 'FAILURE') {
          cobertura coberturaReportFile: '**/coverage.xml', conditionalCoverageTargets: '100,0,80', lineCoverageTargets: '100,0,90'
        }
      }
    }

    // Etapa de Análisis Estático (Flake8)
    stage('Static') {
      steps {
        bat '''
          C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\Scripts\\flake8.exe --exit-zero --format=pylint --exit-zero app
>flake8.out
        '''
        // Umbrales para Flake8
        recordIssues tools: [flake8(name: 'Flake8', pattern: '**/flake8.out')],
                qualityGates: [
                  [threshold: 8, type: 'TOTAL', unstable: true],  // 8 o más hallazgos -> Unstable
                  [threshold: 10, type: 'TOTAL', unstable: false, healthy: false]  // 10 o más hallazgos -> Unhealthy
                ]
      }
    }

    // Etapa de Análisis de Seguridad (Bandit)
    stage('Security') {
      steps {
        bat '''
```

```
            C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\Scripts\\bandit.exe --exit-zero -r . -f custom -o bandit.out --msg-
template "{abspath}:{line}: [{test_id}] {msg}"
        '''
        catchError(buildResult: 'SUCCESS', stageResult: 'UNSTABLE') {
          // Usamos el patrón relativo para buscar 'bandit.out' y aplicar los Quality Gates
          recordIssues tools: [pyLint(name: 'Bandit', pattern: '**/bandit.out')],
                  qualityGates: [
                    [threshold: 2, type: 'TOTAL', unstable: true],   // 2 o más hallazgos -> Unstable
                    [threshold: 4, type: 'TOTAL', unstable: false, healthy: false]  // 4 o más hallazgos -> Unhealthy
                  ]
        }
      }
    }

    // Etapa de Pruebas de Rendimiento (JMeter)
    stage('Performance') {
      steps {
        bat '''
          SET FLASK_APP=app\\api.py
          start /B C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\python.exe -m flask run --host=0.0.0.0 --port=5000

            timeout /t 10 /nobreak
            rem Espera 10 segundos para asegurarse de que Flask esté listo antes de correr las pruebas

          C:\\Users\\amaro\\Downloads\\apache-jmeter-5.6.3\\apache-jmeter-5.6.3\\bin\\jmeter -n -t test\\jmeter\\flask.jmx -f -l flask.jtl
        '''

        script {
          // Verificar cobertura por líneas y ramas (estos valores deben provenir de la herramienta de cobertura)
          def coberturaLineas = 90 // Este valor debe ser obtenido de la herramienta de cobertura
          def coberturaRamas = 85  // Este valor también debe ser obtenido de la herramienta de cobertura

          // Verificar y asignar el estado para cobertura por líneas
          if (coberturaLineas < 85) {
            currentBuild.result = 'FAILURE'
          } else if (coberturaLineas < 95) {
            currentBuild.result = 'UNSTABLE'
          } else {
            currentBuild.result = 'SUCCESS'
          }

          // Verificar y asignar el estado para cobertura por ramas/condiciones
          if (coberturaRamas < 80) {
            currentBuild.result = 'FAILURE'
          } else if (coberturaRamas < 90) {
            currentBuild.result = 'UNSTABLE'
          } else {
            currentBuild.result = 'SUCCESS'
          }

          // Después de la ejecución de las pruebas de rendimiento y la cobertura, procesamos el reporte
          catchError(buildResult: 'SUCCESS', stageResult: 'UNSTABLE') {
            // Ejecutar el reporte de rendimiento
            perfReport sourceDataFiles: '**/flask.jtl'
          }
        }
      }
    }
  }
}
```

Log de la ejecución del pipeline.

```
Lanzada por el usuario Giovanna leon
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Get Code)
[Pipeline] git
The recommended git tool is: NONE
```

```
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/GiovannaLeon/helloworld.git # timeout=10
Fetching upstream changes from https://github.com/GiovannaLeon/helloworld.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.47.1.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/GiovannaLeon/helloworld.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision eb8d8309fcd89b880e7bc330d180dfeceaf53b9a (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe branch -D master # timeout=10
 > git.exe checkout -b master eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
Commit message: "Add files via upload"
 > git.exe rev-list --no-walk eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1>dir
 El volumen de la unidad C es Windows
 El n£mero de serie del volumen es: 6475-04AA

 Directorio de C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1

19/01/2025  23:35   <DIR>        .
19/01/2025  01:21   <DIR>        ..
19/01/2025  23:35         53.248 .coverage
19/01/2025  01:20             40 .gitignore
19/01/2025  01:20   <DIR>        .pytest_cache
19/01/2025  01:20   <DIR>        app
19/01/2025  23:35            378 bandit.out
19/01/2025  23:35          2.237 coverage.xml
19/01/2025  23:35            500 flake8.out
19/01/2025  23:35         19.938 flask.jtl
19/01/2025  01:20   <DIR>        jenkinsFile_1
19/01/2025  01:20   <DIR>        jenkinsFile_2
19/01/2025  01:20   <DIR>        JenkinsFile_3
19/01/2025  01:20   <DIR>        jenkinsfile_4
19/01/2025  23:35         10.397 jmeter.log
19/01/2025  01:20            175 pytest.ini
19/01/2025  01:20            418 README.md
19/01/2025  23:35          1.401 result-unit.xml
19/01/2025  01:20   <DIR>        test
19/01/2025  01:22             28 unit_tests_done.lock
              11 archivos        88.760 bytes
               9 dirs  564.474.511.360 bytes libres
[Pipeline] echo
C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Unit)
[Pipeline] catchError
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1>SET PYTHONPATH=C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1

C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1>C:\Users\amaro\AppData\Local\Programs\Python\Python313\python.exe -m pytest
--junitxml=result-unit.xml test\unit
============================= test session starts =============================
platform win32 -- Python 3.13.0, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1
configfile: pytest.ini
collected 10 items

test\unit\calc_test.py .....                                             [ 80%]
test\unit\util_test.py ..                                                [100%]

- generated xml file: C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1\result-unit.xml -
============================== 10 passed in 0.10s ==============================
[Pipeline] sleep
```

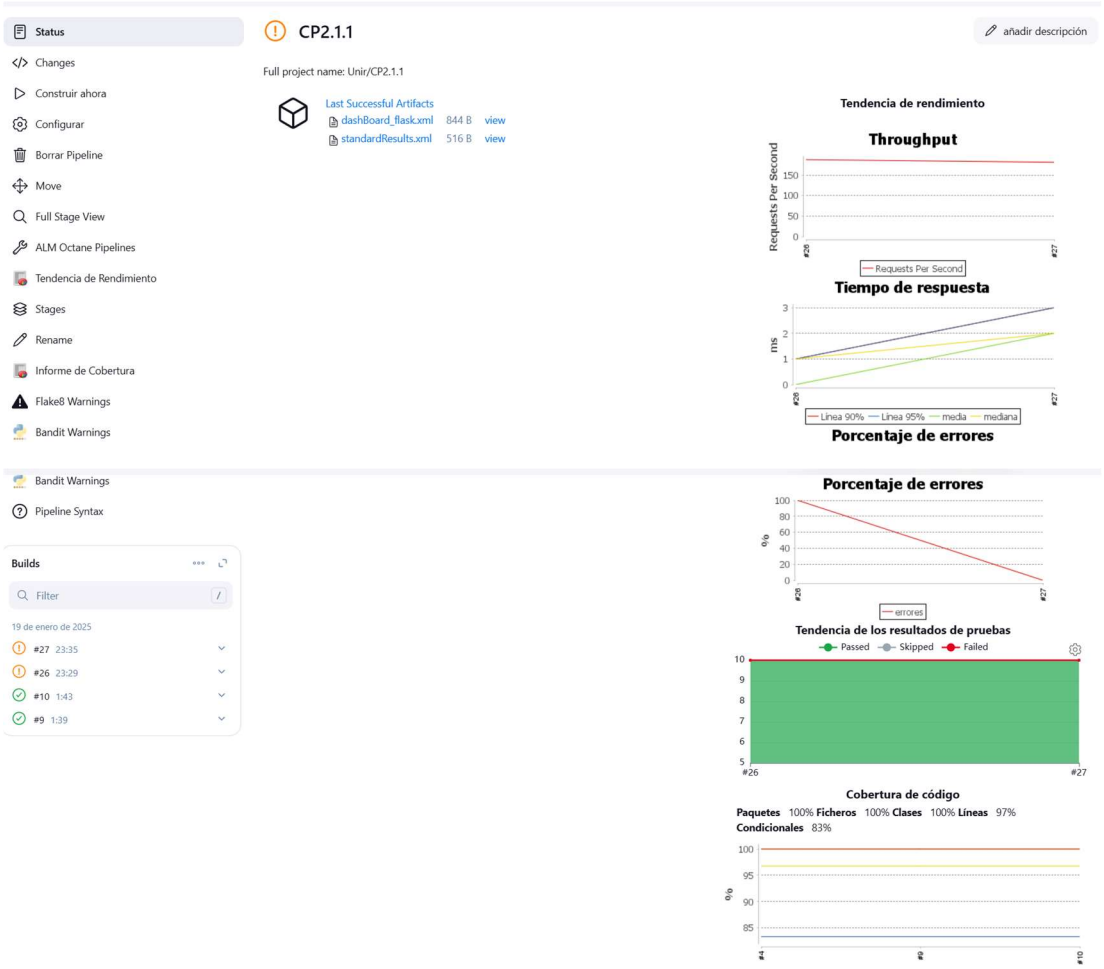
```
Sleeping for 5 Seg
[Pipeline] junit
Grabando resultados de tests
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Coverage)
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1>C:\Users\amaro\AppData\Local\Programs\Python\Python313\Scripts\coverage.exe
run --branch --source=app --omit=app\__init__.py,app\api.py -m pytest test\unit
============================= test session starts =============================
platform win32 -- Python 3.13.0, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1
configfile: pytest.ini
collected 10 items

test\unit\calc_test.py ........                                          [ 80%]
test\unit\util_test.py ..                                                [100%]

============================== 10 passed in 0.03s ==============================

C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1>C:\Users\amaro\AppData\Local\Programs\Python\Python313\Scripts\coverage.exe
xml
Wrote XML report to coverage.xml
[Pipeline] catchError
[Pipeline] {
[Pipeline] cobertura
[Cobertura] Publishing Cobertura coverage report...

[Cobertura] Publishing Cobertura coverage results...

[Cobertura] Cobertura coverage report found.

[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Static)
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1>C:\Users\amaro\AppData\Local\Programs\Python\Python313\Scripts\flake8.exe    --
exit-zero --format=pylint --exit-zero app  1>flake8.out
[Pipeline] recordIssues
WARNING: Unknown parameter(s) found for class type 'io.jenkins.plugins.analysis.core.util.WarningsQualityGate': healthy
[Flake8] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1' that match the pattern '**/flake8.out'
[Flake8] Traversing of symbolic links: enabled
[Flake8] -> found 1 file
[Flake8] Successfully parsed file C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1\flake8.out
[Flake8] -> found 9 issues (skipped 0 duplicates)
[Flake8] Successfully processed file 'flake8.out'
[Flake8] Post processing issues on 'Master' with source code encoding 'windows-1252'
[Flake8] Creating SCM blamer to obtain author and commit information for affected files
[Flake8] -> No blamer installed yet. You need to install the 'git-forensics' plugin to enable blaming for Git.
[Flake8] Resolving file names for all issues in workspace 'C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1'
[Flake8] -> resolved paths in source directory (2 found, 0 not found)
[Flake8] Resolving module names from module definitions (build.xml, pom.xml, or Manifest.mf files)
[Flake8] -> resolved module names for 9 issues
[Flake8] Resolving package names (or namespaces) by parsing the affected files
[Flake8] -> resolved package names of 2 affected files
[Flake8] No filter has been set, publishing all 9 issues
[Flake8] Creating fingerprints for all affected code blocks to track issues over different builds
[Flake8] -> created fingerprints for 9 issues (skipped 0 issues)
[Flake8] Copying affected files to Jenkins' build folder 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\CP2.1.1\builds\28\files-with-issues'
[Flake8] -> 2 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[Flake8] Skipping cleaning of source code files in old builds
[Flake8] Repository miner is not configured, skipping repository mining
[Flake8] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1' that match the pattern '**/flake8.out'
[Flake8] Traversing of symbolic links: enabled
```

```
[Flake8] -> found 1 file
[Flake8] Successfully parsed file C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1\flake8.out
[Flake8] -> found 9 issues (skipped 0 duplicates)
[Flake8] Successfully processed file 'flake8.out'
[Flake8] Post processing issues on 'Master' with source code encoding 'windows-1252'
[Flake8] Creating SCM blamer to obtain author and commit information for affected files
[Flake8] -> No blamer installed yet. You need to install the 'git-forensics' plugin to enable blaming for Git.
[Flake8] Resolving file names for all issues in workspace 'C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1'
[Flake8] -> resolved paths in source directory (2 found, 0 not found)
[Flake8] Resolving module names from module definitions (build.xml, pom.xml, or Manifest.mf files)
[Flake8] -> resolved module names for 9 issues
[Flake8] Resolving package names (or namespaces) by parsing the affected files
[Flake8] -> resolved package names of 2 affected files
[Flake8] No filter has been set, publishing all 9 issues
[Flake8] Creating fingerprints for all affected code blocks to track issues over different builds
[Flake8] -> created fingerprints for 9 issues (skipped 0 issues)
[Flake8] Copying affected files to Jenkins' build folder 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\CP2.1.1\builds\28\files-with-issues'
[Flake8] -> 2 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[Flake8] Skipping cleaning of source code files in old builds
[Flake8] Repository miner is not configured, skipping repository mining
[Flake8] Reference build recorder is not configured
[Flake8] No valid reference build found
[Flake8] All reported issues will be considered outstanding
[Flake8] Evaluating quality gates
[Flake8] -> Some quality gates have been missed: overall result is UNSTABLE
[Flake8] -> Details for each quality gate:
[Flake8]    - [Total (any severity)]: ≪Inestable≫ - (Actual value: 9, Quality gate: 8,00)
[Flake8]    - [Total (any severity)]: ≪Correcto≫ - (Actual value: 9, Quality gate: 10,00)
[Flake8] Health report is disabled - skipping
[Flake8] Created analysis result for 9 issues (found 0 new issues, fixed 0 issues)
[Flake8] Attaching ResultAction with ID 'flake8' to build 'Unir/CP2.1.1 #28'.
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Security)
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1>C:\Users\amaro\AppData\Local\Programs\Python\Python313\Scripts\bandit.exe  --
exit-zero -r . -f custom -o bandit.out --msg-template "{abspath}:{line}: [{test_id}] {msg}"
[main]      INFO        profile include tests: None
[main]      INFO        profile exclude tests: None
[main]      INFO        cli include tests: None
[main]      INFO        cli exclude tests: None
[main]      INFO        running on Python 3.13.0
[custom]    INFO        Result written to file: bandit.out
[Pipeline] catchError
[Pipeline] {
[Pipeline] recordIssues
WARNING: Unknown parameter(s) found for class type 'io.jenkins.plugins.analysis.core.util.WarningsQualityGate': healthy
[Bandit] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1' that match the pattern '**/bandit.out'
[Bandit] Traversing of symbolic links: enabled
[Bandit] -> found 1 file
[Bandit] Successfully parsed file C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1\bandit.out
[Bandit] -> found 2 issues (skipped 0 duplicates)
[Bandit] Successfully processed file 'bandit.out'
[Bandit] Post processing issues on 'Master' with source code encoding 'windows-1252'
[Bandit] Creating SCM blamer to obtain author and commit information for affected files
[Bandit] -> No blamer installed yet. You need to install the 'git-forensics' plugin to enable blaming for Git.
[Bandit] Resolving file names for all issues in workspace 'C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1'
[Bandit] -> resolved paths in source directory (1 found, 0 not found)
[Bandit] Resolving module names from module definitions (build.xml, pom.xml, or Manifest.mf files)
[Bandit] -> resolved module names for 2 issues
[Bandit] Resolving package names (or namespaces) by parsing the affected files
[Bandit] -> resolved package names of 1 affected files
[Bandit] No filter has been set, publishing all 2 issues
[Bandit] Creating fingerprints for all affected code blocks to track issues over different builds
[Bandit] -> created fingerprints for 2 issues (skipped 0 issues)
[Bandit] Copying affected files to Jenkins' build folder 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\CP2.1.1\builds\28\files-with-issues'
[Bandit] -> 1 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[Bandit] Skipping cleaning of source code files in old builds
[Bandit] Repository miner is not configured, skipping repository mining
[Bandit] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1' that match the pattern '**/bandit.out'
```

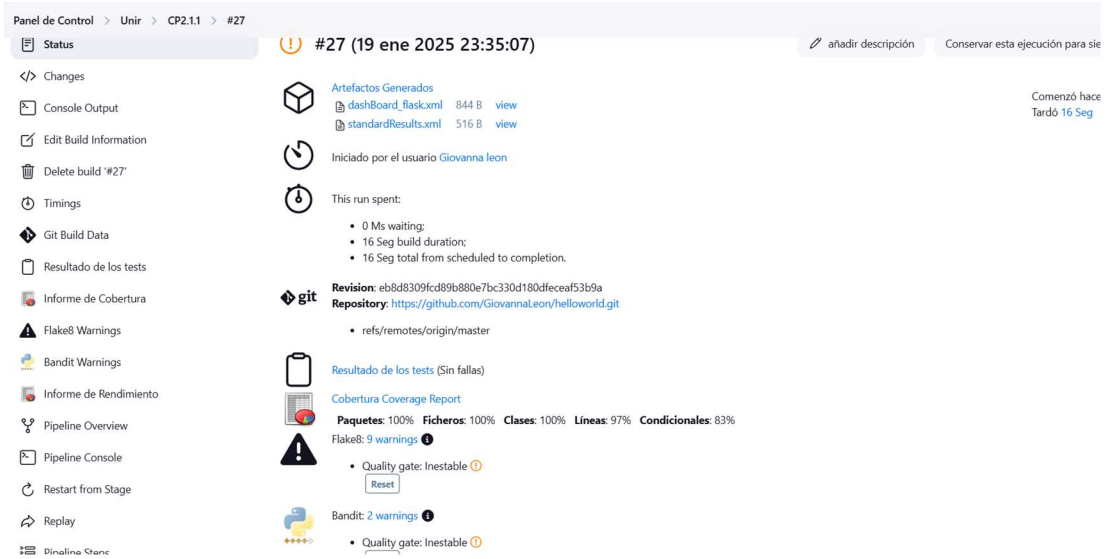
```
[Bandit] Traversing of symbolic links: enabled
[Bandit] -> found 1 file
[Bandit] Successfully parsed file C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1\bandit.out
[Bandit] -> found 2 issues (skipped 0 duplicates)
[Bandit] Successfully processed file 'bandit.out'
[Bandit] Post processing issues on 'Master' with source code encoding 'windows-1252'
[Bandit] Creating SCM blamer to obtain author and commit information for affected files
[Bandit] -> No blamer installed yet. You need to install the 'git-forensics' plugin to enable blaming for Git.
[Bandit] Resolving file names for all issues in workspace 'C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1'
[Bandit] -> resolved paths in source directory (1 found, 0 not found)
[Bandit] Resolving module names from module definitions (build.xml, pom.xml, or Manifest.mf files)
[Bandit] -> resolved module names for 2 issues
[Bandit] Resolving package names (or namespaces) by parsing the affected files
[Bandit] -> resolved package names of 1 affected files
[Bandit] No filter has been set, publishing all 2 issues
[Bandit] Creating fingerprints for all affected code blocks to track issues over different builds
[Bandit] -> created fingerprints for 2 issues (skipped 0 issues)
[Bandit] Copying affected files to Jenkins' build folder 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\CP2.1.1\builds\28\files-with-issues'
[Bandit] -> 1 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[Bandit] Skipping cleaning of source code files in old builds
[Bandit] Repository miner is not configured, skipping repository mining
[Bandit] Reference build recorder is not configured
[Bandit] No valid reference build found
[Bandit] All reported issues will be considered outstanding
[Bandit] Evaluating quality gates
[Bandit] -> Some quality gates have been missed: overall result is UNSTABLE
[Bandit] -> Details for each quality gate:
[Bandit]    - [Total (any severity)]: ≪Inestable≫ - (Actual value: 2, Quality gate: 2,00)
[Bandit]    - [Total (any severity)]: ≪Correcto≫ - (Actual value: 2, Quality gate: 4,00)
[Bandit] Health report is disabled - skipping
[Bandit] Created analysis result for 2 issues (found 0 new issues, fixed 0 issues)
[Bandit] Attaching ResultAction with ID 'pylint' to build 'Unir/CP2.1.1 #28'.
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Performance)
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1>SET FLASK_APP=app\api.py

C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1>start  /B C:\Users\amaro\AppData\Local\Programs\Python\Python313\python.exe -
m flask run --host=0.0.0.0 --port=5000

C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1>C:\Users\amaro\Downloads\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\jmeter  -
n -t test\jmeter\flask.jmx -f -l flask.jtl
 * Serving Flask app 'app\api.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.1.48:5000
Press CTRL+C to quit
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
Creating summariser <summary>
Created the tree successfully using test\jmeter\flask.jmx
Starting standalone test @ 2025 Jan 20 01:04:47 CET (1737331487279)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary =    160 in 00:00:01 =  163.3/s Avg:     2 Min:     1 Max:    26 Err:     0 (0.00%)
Tidying up ...    @ 2025 Jan 20 01:04:48 CET (1737331488341)
... end of run
[Pipeline] script
[Pipeline] {
[Pipeline] catchError
[Pipeline] {
[Pipeline] perfReport
Creating parser with percentiles:'0,50,90,95,100,' filterRegex:null
Performance: Recording JMeterCsv reports '**/flask.jtl'
```

Performance: JMeterCsv copying reports to master, files '[C:\ProgramData\Jenkins\.jenkins\workspace\Unir\CP2.1.1\flask.jtl]'
Performance: JMeterCsv parsing local reports '[C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\CP2.1.1\builds\28\performance-reports\JMeterCSV\flask.jtl]'
Performance: Parsing report file 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\CP2.1.1\builds\28\performance-reports\JMeterCSV\flask.jtl' with filterRegex 'null'.
Performance: No threshold configured for making the test unstable
Performance: No threshold configured for making the test failure
Performance: File flask.jtl reported 0.0% of errors [SUCCESS]. Build status is: UNSTABLE
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: UNSTABLE

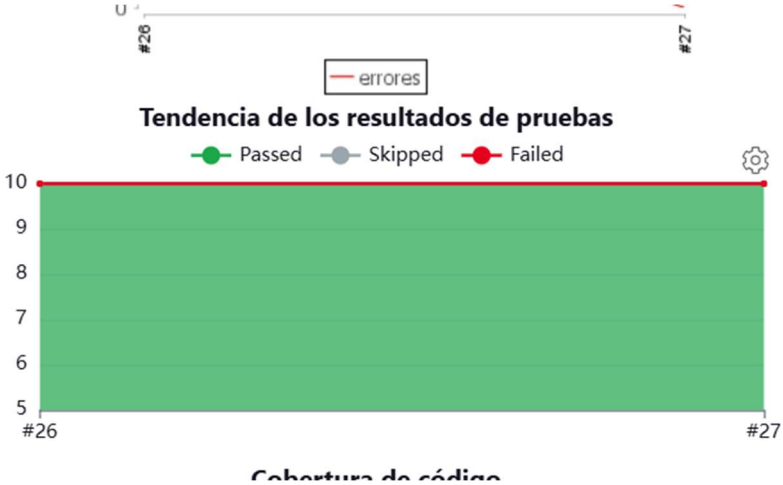
- Captura de pantalla de Jenkins donde se vea el resultado de la ejecución del pipeline.

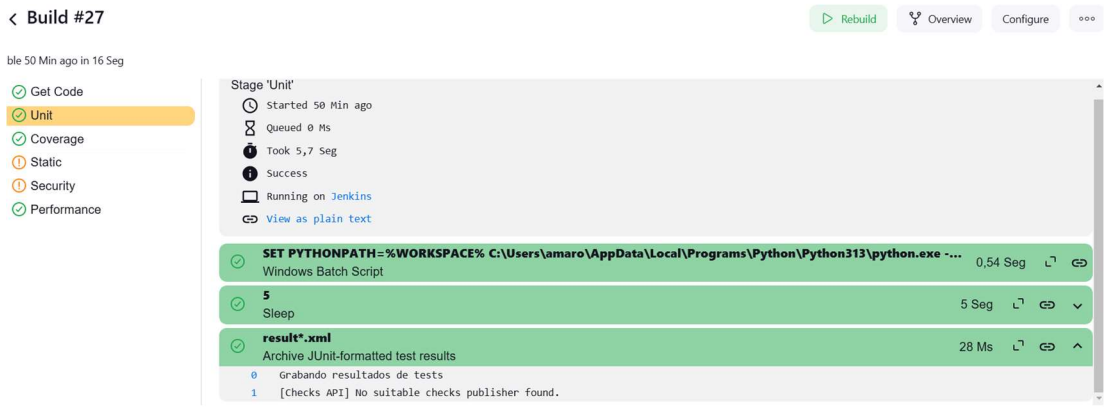


Resultados

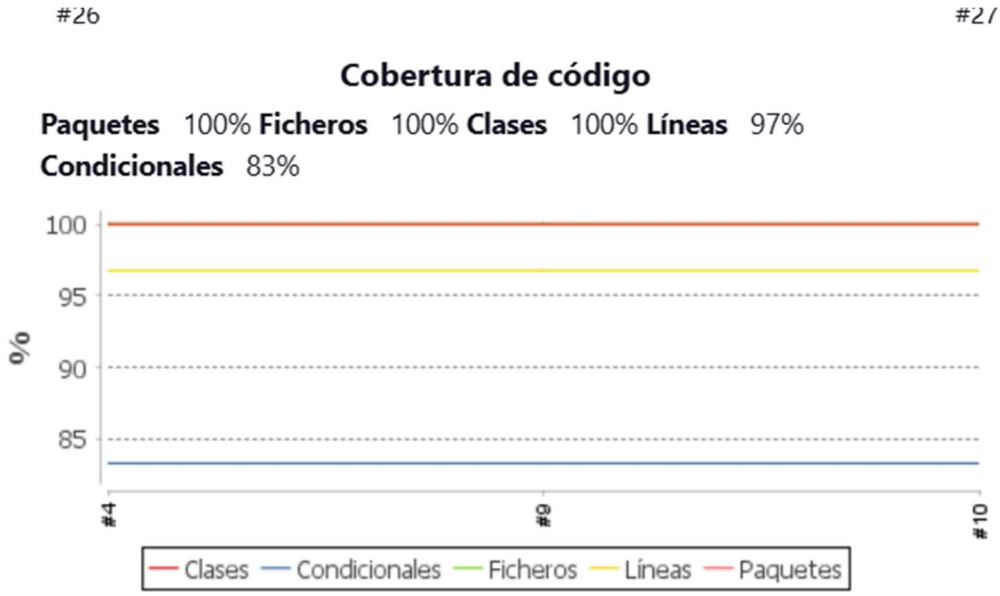


- Captura de pantalla de Jenkins donde se vea el resultado de los plugins:
  - Plugin Junit: evolución de los resultados de las pruebas unitarias.



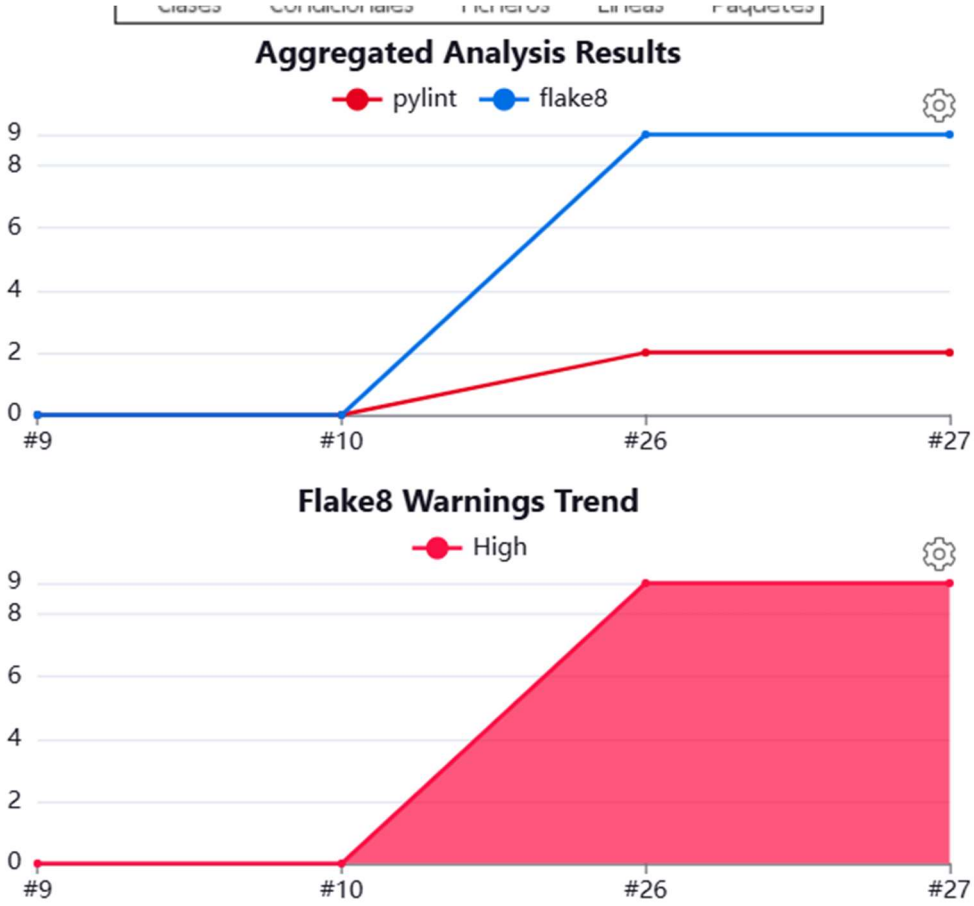


- Plugin Cobertura: evolución de la cobertura por líneas, ramas, etc.

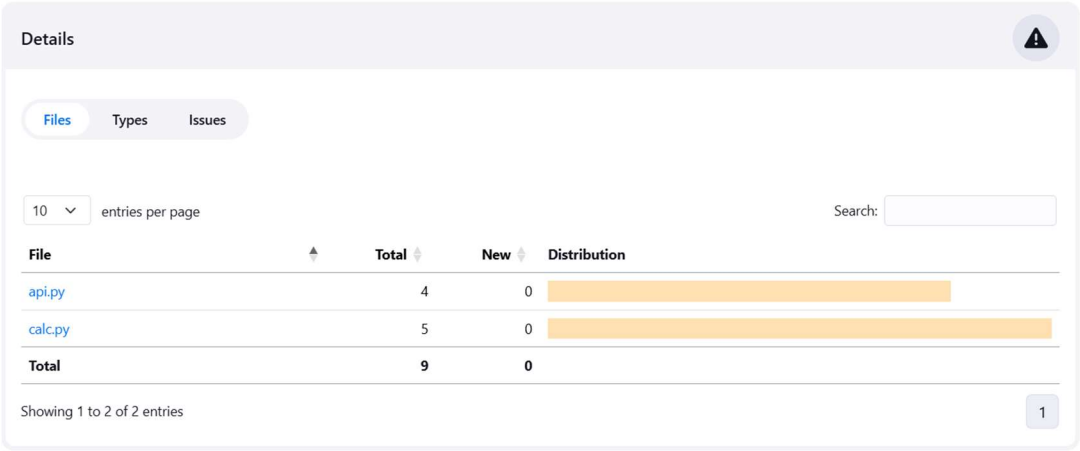


- 

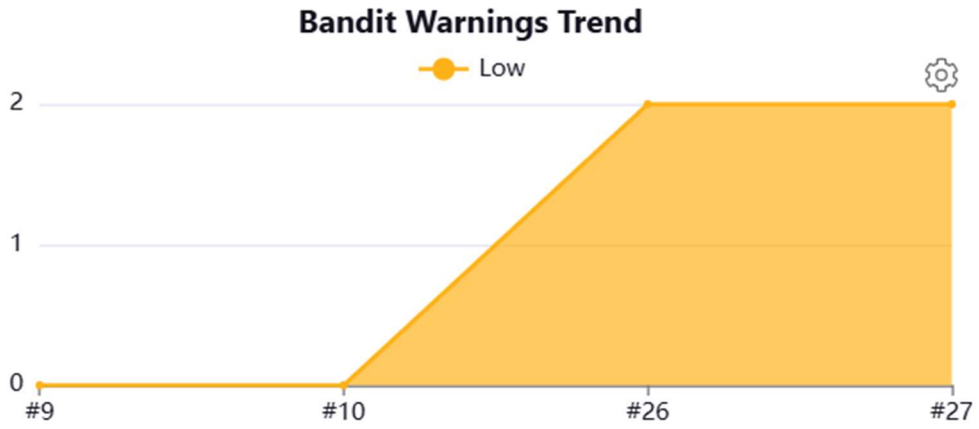
- Plugin Warnings-NG (Flake8): evolución de los hallazgos encontrados.

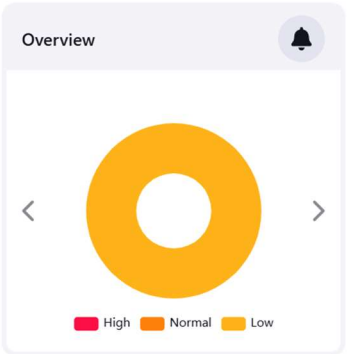


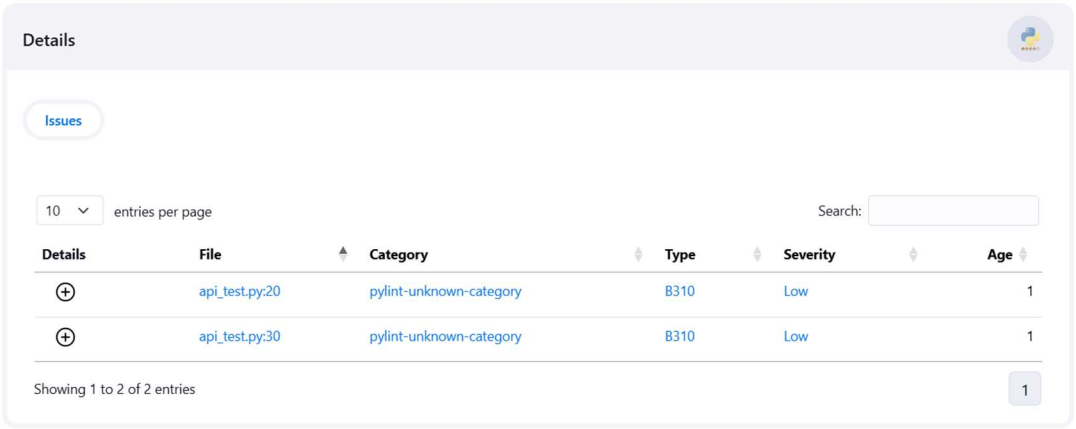
Flake8 Warnings



- Plugin Warnings-NG (Bandit): evolución de los hallazgos encontrados.







Details

Issues

10 entries per page

Search:

Details	File	Category	Type	Severity	Age
⊕	api_test.py:20	pylint-unknown-category	B310	Low	1
⊕	api_test.py:30	pylint-unknown-category	B310	Low	1

Showing 1 to 2 of 2 entries

1

- Plugin Performance: gráfica de tiempos de respuesta “Response Times (ms)” de ambos microservicios, y comentario sobre qué conclusiones se pueden extraer al observar esa gráfica.
  - Obtener el valor (aproximado) de línea 90 para el microservicio de suma e indicar en qué gráfica se puede obtener este dato, y qué significa este dato.

Status
Changes
Console Output
Edit Build Information
Delete build '#28'
Timings
Git Build Data

Rendimiento por URI: flask.jtl

Response time trends for build: "Unir/CP2.1.1 #28"

Comparison with previous build

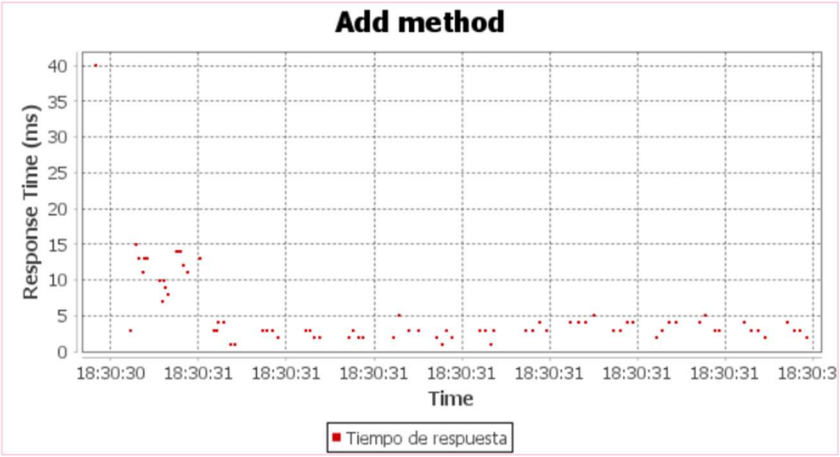
URI	Samples	Media (ms)	Min(ms)	Median(ms)	Line 90.0(ms)	Line 95.0(ms)	Max(ms)	Http Code	Errores (%)	Media (KB)	Total (KB)
Add method	80 +0	2 0	1 0	2 0	3 0	4 0	26 +2	200	0.0 % 0.0 %	0.31 0.0	24.69 0.0
Substract method	80 +0	2 +1	1 0	2 0	3 0	4 +1	5 0	200	0.0 % 0.0 %	0.32 0.0	25.23 0.0
Todas las URIs	**160 +0**	**2 0**	**1 0**	**2 0**	**3 0**	**4 +1**	**26 +2**		**0.0 % 0.0 %**	**0.31**	**49.92**

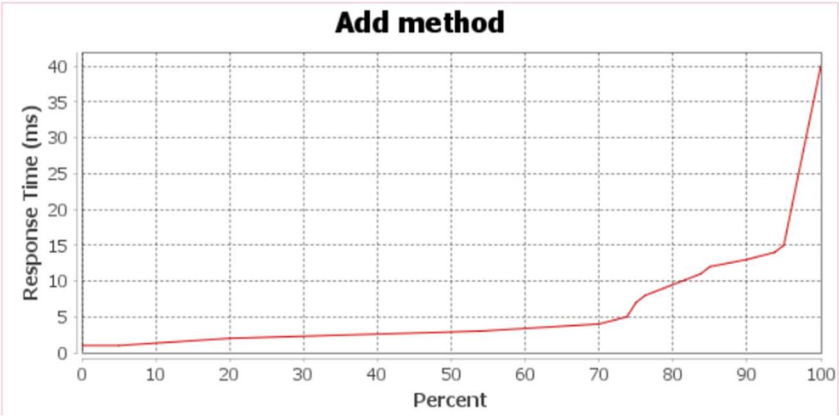
URI: Add method

URI	Samples	Media (ms)	Min(ms)	Median(ms)	Line 90.0(ms)	Line 95.0(ms)	Max(ms)	Http Code	Errores (%)	Media (KB)	Total (KB)
Add method	80 +0	2 0	1 0	2 0	3 0	4 0	26 +2	200	0.0 % 0.0 %	0.31 0.0	24.69 0.0

Peticiones

Http Code	Fecha	Duración (ms)
200	Mon Jan 20 01:04:47 CET 2025	26 ms.
200	Mon Jan 20 01:04:47 CET 2025	10 ms.
200	Mon Jan 20 01:04:47 CET 2025	26 ms.
200	Mon Jan 20 01:04:47 CET 2025	4 ms.
200	Mon Jan 20 01:04:47 CET 2025	5 ms.
200	Mon Jan 20 01:04:47 CET 2025	4 ms.
200	Mon Jan 20 01:04:47 CET 2025	3 ms.
200	Mon Jan 20 01:04:47 CET 2025	2 ms.
200	Mon Jan 20 01:04:47 CET 2025	3 ms.
200	Mon Jan 20 01:04:47 CET 2025	4 ms.
200	Mon Jan 20 01:04:47 CET 2025	3 ms.
200	Mon Jan 20 01:04:47 CET 2025	4 ms.
200	Mon Jan 20 01:04:47 CET 2025	3 ms.
200	Mon Jan 20 01:04:47 CET 2025	2 ms.
200	Mon Jan 20 01:04:47 CET 2025	2 ms.
200	Mon Jan 20 01:04:47 CET 2025	2 ms.
200	Mon Jan 20 01:04:47 CET 2025	2 ms.
200	Mon Jan 20 01:04:47 CET 2025	1 ms.
200	Mon Jan 20 01:04:47 CET 2025	2 ms.
200	Mon Jan 20 01:04:47 CET 2025	1 ms.
200	Mon Jan 20 01:04:47 CET 2025	2 ms.
200	Mon Jan 20 01:04:47 CET 2025	2 ms.
200	Mon Jan 20 01:04:47 CET 2025	2 ms.





Significa: Que en la línea 90 suele tardar menos de 3 ms.

- Explicación del funcionamiento del pipeline.

Este **Jenkins Pipeline** está diseñado para realizar una serie de acciones relacionadas con el ciclo de vida de un proyecto de desarrollo de software. Aquí explico:

Estructura General

Este pipeline está dividido en varias **etapas** (stages), cada una responsable de realizar diferentes tipos de validaciones y pruebas en el código fuente. Cada etapa puede definir su propio comportamiento, y si alguna etapa falla, el pipeline puede continuar o terminar dependiendo de la configuración.

1. Etapa 'Get Code'

- **Propósito:** Obtener el código fuente del repositorio de GitHub y verificar el entorno de trabajo.
- **Pasos:**
    - Se realiza un **git clone** para obtener el código del repositorio público de GitHub: https://github.com/GiovannaLeon/helloworld.git.

- Luego, se ejecuta el comando **bat "dir"** para listar los archivos en el directorio de trabajo.
- Finalmente, se imprime el valor de la variable WORKSPACE, que es el directorio en el que Jenkins está ejecutando el pipeline.

2. Etapa 'Unit' (Pruebas Unitarias)

- **Propósito:** Ejecutar las pruebas unitarias utilizando pytest y generar un informe de resultados.
- **Pasos:**
  - Utiliza **catchError** para atrapar errores. Si ocurre un error, el pipeline continuará, pero se marcará como **UNSTABLE**.
  - Se ejecutan las pruebas unitarias usando **pytest**, y los resultados se guardan en un archivo XML (result-unit.xml).
  - Después de ejecutar las pruebas, se espera 5 segundos con **sleep(5)**.
  - Se genera un informe de pruebas unitarias usando **junit** con los archivos result*.xml.

3. Etapa 'Coverage' (Cobertura de Pruebas)

- **Propósito:** Calcular la cobertura del código fuente mediante la herramienta coverage.
- **Pasos:**
  - Se ejecuta el comando coverage para calcular la cobertura de pruebas, incluyendo las ramas y excluyendo ciertos archivos (app__init__.py, app\\api.py).
  - Después de ejecutar coverage, se genera un archivo coverage.xml.
  - Se usa **catchError** para manejar cualquier error, permitiendo que el pipeline continúe, pero marcando la etapa como **UNSTABLE** en caso de error.
  - Se ejecuta el plugin **cobertura** para mostrar el reporte de cobertura, especificando umbrales para la cobertura de líneas y condiciones de cobertura de ramas.

4. Etapa 'Static' (Análisis Estático con Flake8)

- **Propósito:** Ejecutar un análisis estático del código para identificar problemas de estilo y calidad utilizando flake8.
- **Pasos:**
  - Se ejecuta **flake8** con formato pylint, analizando los archivos del directorio app y guardando los resultados en el archivo flake8.out.
  - Se utiliza el plugin **recordIssues** para registrar los hallazgos de flake8 y establecer los umbrales de calidad:
    - Si hay **8 o más hallazgos**, el pipeline se marcará como **UNSTABLE**.
    - Si hay **10 o más hallazgos**, el pipeline se marcará como **UNHEALTHY**.

5. Etapa 'Security' (Análisis de Seguridad con Bandit)

- **Propósito:** Realizar un análisis de seguridad con la herramienta bandit para identificar vulnerabilidades potenciales.
- **Pasos:**
    - Se ejecuta **bandit** en el proyecto para buscar vulnerabilidades de seguridad. Los resultados se guardan en el archivo bandit.out.
    - Utiliza **catchError** para que, si se produce un error, la etapa continúe con el estado **UNSTABLE**.
    - Al igual que en la etapa de análisis estático, se usa **recordIssues** para registrar los hallazgos de seguridad y establecer umbrales:
        - Si hay **2 o más hallazgos**, el pipeline se marcará como **UNSTABLE**.
        - Si hay **4 o más hallazgos**, el pipeline se marcará como **UNHEALTHY**.

6. Etapa 'Performance' (Pruebas de Rendimiento con JMeter)

- **Propósito:** Ejecutar pruebas de rendimiento usando **JMeter** y evaluar el desempeño de la aplicación.
- **Pasos:**
    - Se configura el entorno de **Flask** estableciendo la variable de entorno FLASK_APP y se inicia el servidor Flask en segundo plano.
    - Luego, se ejecuta **JMeter** usando un archivo de prueba .jmx que está ubicado en test\\jmeter\\flask.jmx. Los resultados de las pruebas de rendimiento se guardan en flask.jtl.
    - En el bloque **script**, se define un código para verificar la cobertura de líneas y ramas:
        - Si la cobertura de líneas es menor a **85**, el estado de la construcción se marca como **FAILURE**.
        - Si la cobertura de líneas es entre **85 y 95**, se marca como **UNSTABLE**.
        - Si la cobertura de líneas es mayor a **95**, se marca como **SUCCESS**.
        - Lo mismo se hace para la cobertura de ramas con umbrales similares.
    - Finalmente, se ejecuta el reporte de rendimiento de **JMeter** usando el plugin **perfReport**.

Resumen de los comportamientos:

1. **catchError:** Utilizado en varias etapas (Unit, Coverage, Security) para permitir que el pipeline continúe incluso si hay errores, pero marcando esas etapas como **UNSTABLE** si ocurre un error.

2. **recordIssues:** Se usa en las etapas de análisis estático y de seguridad para aplicar umbrales de calidad. Si el número de hallazgos supera los umbrales, el estado de la etapa se marca como **UNSTABLE** o **UNHEALTHY**.

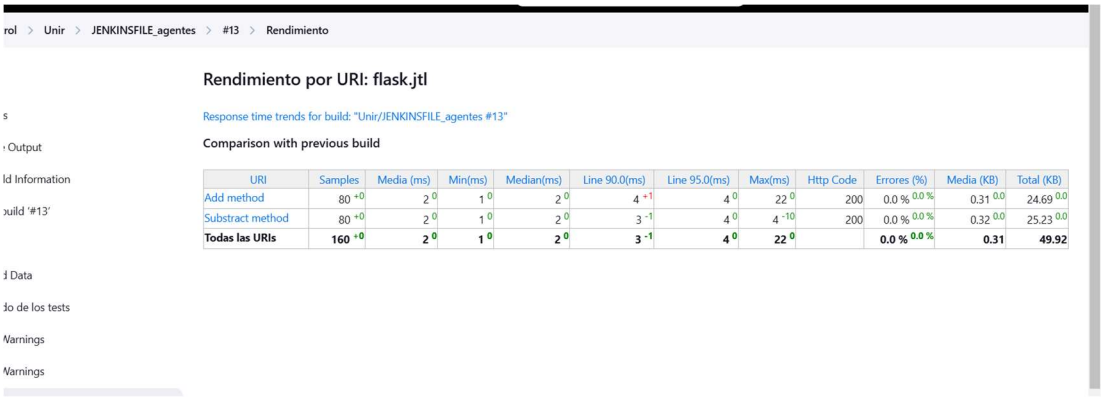
3. **Cobertura de pruebas y rendimiento:** Si las métricas de cobertura de código (líneas y ramas) son bajas, el pipeline se marca como **UNSTABLE** o **FAILURE**.

Este pipeline cubre múltiples aspectos importantes del desarrollo de software como pruebas unitarias, cobertura de código, calidad estática, análisis de seguridad y pruebas de rendimiento.

Reto 2 – Distribución en agentes

En este reto se solicitan 3 entregables:

- URL al repositorio creado por el alumno, a partir del código fuente base de este CP1, que albergue tanto el código fuente como el nuevo Jenkinsfile.



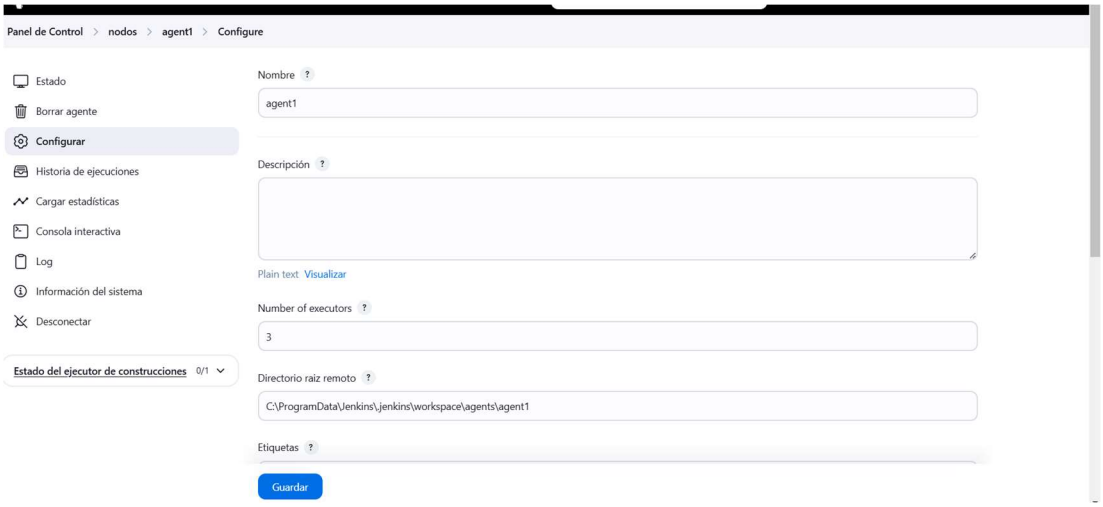
URI	Samples	Media (ms)	Min(ms)	Median(ms)	Line 90.0(ms)	Line 95.0(ms)	Max(ms)	Http Code	Errores (%)	Media (KB)	Total (KB)
Add method	80 +0	2 0	1 0	2 0	4 +1	4 0	22 0	200	0.0 % 0.0 %	0.31 0.0	24.69 0.0
Substract method	80 +0	2 0	1 0	2 0	3 -1	4 0	4 -10	200	0.0 % 0.0 %	0.32 0.0	25.23 0.0
Todas las URIs	**160 +0**	**2 0**	**1 0**	**2 0**	**3 -1**	**4 0**	**22 0**		**0.0 % 0.0 %**	**0.31**	**49.92**

```
    agent none

    stages {
        stage('Get Code') {
            agent { label 'agent1' }  // Agente principal
            steps {
                bat "whoami"
                bat "hostname"
                bat "echo ${WORKSPACE}"
                git 'https://github.com/GiovannaLeon/helloworld.git'
                bat "dir"
                echo "Workspace: ${WORKSPACE}"
            }
        }

        stage('Unit') {
            agent { label 'agent1' }  // Agente principal
            steps {
                bat "whoami"
                bat "hostname"
                bat "echo ${WORKSPACE}"
                catchError(buildResult: 'UNSTABLE', stageResult: 'FAILURE') {
                    bat '''
                        SET PYTHONPATH=%WORKSPACE%
                        C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\python.exe -m pytest --junitxml=result-unit.xml test\\unit
                    '''
                    sleep(5)
                    junit 'result*.xml' // Reporte de las pruebas unitarias
                }
            }
        }

        stage('Static and Security Analysis') {
            parallel {
                stage('Static') {
                    agent { label 'agent1' }  // Agente principal
                    steps {
                        bat "whoami"
                        bat "hostname"
                        bat "echo ${WORKSPACE}"
                        bat '''
                            C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\Scripts\\flake8.exe  --exit-zero  --format=pylint  --exit-zero
app >flake8.out
                        '''
                        // Umbrales para Flake8
                        recordIssues tools: [flake8(name: 'Flake8', pattern: '**/flake8.out')],
                                qualityGates: [
                                    [threshold: 8, type: 'TOTAL', unstable: true],  // 8 o más hallazgos -> Unstable
                                    [threshold: 10, type: 'TOTAL', unstable: false, healthy: false]  // 10 o más hallazgos -> Unhealthy
                                ]
                    }
                }

                stage('Security') {
```

```
            agent { label 'agent2' }  // Agente dedicado a seguridad
            steps {
               bat "whoami"
               bat "hostname"
               bat "echo ${WORKSPACE}"
               bat '''
                  C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\Scripts\\bandit.exe --exit-zero -r . -f custom -o bandit.out --msg-template "{abspath}:{line}: [{test_id}] {msg}"
               '''
               catchError(buildResult: 'SUCCESS', stageResult: 'UNSTABLE') {
                  // Usamos el patrón relativo para buscar 'bandit.out' y aplicar los Quality Gates
                  recordIssues tools: [pyLint(name: 'Bandit', pattern: '**/bandit.out')],
                           qualityGates: [
                              [threshold: 2, type: 'TOTAL', unstable: true],   // 2 o más hallazgos -> Unstable
                              [threshold: 4, type: 'TOTAL', unstable: false, healthy: false]  // 4 o más hallazgos -> Unhealthy
                           ]
               }
            }
         }
      }
   }

   stage('Coverage') {
      agent { label 'agent1' }  // Agente principal
      steps {
         bat "whoami"
         bat "hostname"
         bat "echo ${WORKSPACE}"
         bat '''
            C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\Scripts\\coverage.exe run --branch --source=app --omit=app\\__init__.py,app\\api.py -m pytest test\\unit
            C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\Scripts\\coverage.exe xml
         '''
         catchError(buildResult: 'UNSTABLE', stageResult: 'FAILURE') {
            cobertura coberturaReportFile: '**/coverage.xml', conditionalCoverageTargets: '100,0,80', lineCoverageTargets: '100,0,90'
         }
      }
   }

  stage('Performance') {
      agent { label 'agent3' }  // Agente dedicado a pruebas de rendimiento
      steps {
         bat "whoami"
         bat "hostname"
         bat "echo ${WORKSPACE}"
         bat '''
          SET FLASK_APP=app\\api.py
            start /B C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\python.exe -m flask run --host=0.0.0.0 --port=5000
              timeout /t 10 /nobreak
             rem Espera 10 segundos para asegurarse de que Flask esté listo antes de correr las pruebas

            C:\\Users\\amaro\\Downloads\\apache-jmeter-5.6.3\\apache-jmeter-5.6.3\\bin\\jmeter -n -t test\\jmeter\\flask.jmx -f -l flask.jtl
         '''
         script {
            // Verificar cobertura por líneas y ramas (estos valores deben provenir de la herramienta de cobertura)
            def coberturaLineas = 90 // Este valor debe ser obtenido de la herramienta de cobertura
            def coberturaRamas = 85  // Este valor también debe ser obtenido de la herramienta de cobertura

            // Verificar y asignar el estado para cobertura por líneas
            if (coberturaLineas < 85) {
               currentBuild.result = 'FAILURE'
            } else if (coberturaLineas < 95) {
               currentBuild.result = 'UNSTABLE'
            } else {
               currentBuild.result = 'SUCCESS'
            }

            // Verificar y asignar el estado para cobertura por ramas/condiciones
            if (coberturaRamas < 80) {
               currentBuild.result = 'FAILURE'
            } else if (coberturaRamas < 90) {
               currentBuild.result = 'UNSTABLE'
            } else {
```

```
                currentBuild.result = 'SUCCESS'
            }

            // Después de la ejecución de las pruebas de rendimiento y la cobertura, procesamos el reporte
            // catchError(buildResult: 'SUCCESS', stageResult: 'UNSTABLE') {
                // Ejecutar el reporte de rendimiento
                perfReport sourceDataFiles: '**/flask.jtl'
            // }
            }
        }
    }

  }
}
```

- Log de la ejecución del pipeline (debe visualizarse un “whoami” y “hostname” para identificar el agente empleado en cada etapa).




```
Lanzada por el usuario Giovanna leon
[Pipeline] Start of Pipeline
[Pipeline] stage
[Pipeline] { (Get Code)
[Pipeline] node
Running on agent1 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Fetching changes from the remote Git repository
Checking out Revision eb8d8309fcd89b880e7bc330d180dfeceaf53b9a (refs/remotes/origin/master)
Commit message: "Add files via upload"
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes\.git # timeout=10
 > git.exe config remote.origin.url https://github.com/GiovannaLeon/helloworld.git # timeout=10
Fetching upstream changes from https://github.com/GiovannaLeon/helloworld.git
```

```
 > git.exe --version # timeout=10
 > git --version # 'git version 2.47.1.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/GiovannaLeon/helloworld.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe branch -D master # timeout=10
 > git.exe checkout -b master eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
 > git.exe rev-list --no-walk eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>dir
 El volumen de la unidad C es Windows
 El n£mero de serie del volumen es: 6475-04AA

 Directorio de C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes

20/01/2025  21:37    <DIR>          .
20/01/2025  01:38    <DIR>          ..
20/01/2025  21:37            53.248 .coverage
20/01/2025  01:38                40 .gitignore
20/01/2025  01:38    <DIR>          .pytest_cache
20/01/2025  01:38    <DIR>          app
20/01/2025  21:37             2.273 coverage.xml
20/01/2025  21:37               500 flake8.out
20/01/2025  01:38    <DIR>          jenkinsFile_1
20/01/2025  01:38    <DIR>          jenkinsFile_2
20/01/2025  01:38    <DIR>          JenkinsFile_3
20/01/2025  01:38    <DIR>          jenkinsfile_4
20/01/2025  01:38               175 pytest.ini
20/01/2025  01:38               418 README.md
20/01/2025  21:37             1.401 result-unit.xml
20/01/2025  01:38    <DIR>          test
               7 archivos         58.055 bytes
               9 dirs  564.940.984.320 bytes libres
[Pipeline] echo
Workspace: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Unit)
[Pipeline] node
Running on agent1 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] catchError
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>SET
PYTHONPATH=C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>C:\Users\amaro\AppData\Local\Programs\
Python\Python313\python.exe -m pytest --junitxml=result-unit.xml test\unit
============================= test session starts =============================
platform win32 -- Python 3.13.0, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
configfile: pytest.ini
```

```
collected 10 items

test\unit\calc_test.py .....                                [ 80%]
test\unit\util_test.py ..                                   [100%]

- generated xml file: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes\result-unit.xml -
============================== 10 passed in 0.10s ==============================
[Pipeline] sleep
Sleeping for 5 Seg
[Pipeline] junit
Grabando resultados de tests
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Static and Security Analysis)
[Pipeline] parallel
[Pipeline] { (Branch: Static)
[Pipeline] { (Branch: Security)
[Pipeline] stage
[Pipeline] { (Static)
[Pipeline] stage
[Pipeline] { (Security)
[Pipeline] node
Running on agent1 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] node
Running on agent2 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes
[Pipeline] {
[Pipeline] {
[Pipeline] bat
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>C:\Users\amaro\AppData\Local\Programs\
Python\Python313\Scripts\flake8.exe --exit-zero --format=pylint --exit-zero app  1>flake8.out

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes>C:\Users\amaro\AppData\Local\Programs\
Python\Python313\Scripts\bandit.exe --exit-zero -r . -f custom -o bandit.out --msg-template "{abspath}:{line}: [{test_id}] {msg}"
[main]      INFO        profile include tests: None
[main]      INFO        profile exclude tests: None
[main]      INFO        cli include tests: None
[main]      INFO        cli exclude tests: None
[main]      INFO        running on Python 3.13.0
[custom]    INFO        Result written to file: bandit.out
```

```
[Pipeline] catchError
[Pipeline] {
[Pipeline] recordIssues
WARNING: Unknown parameter(s) found for class type 'io.jenkins.plugins.analysis.core.util.WarningsQualityGate': healthy
[Bandit] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes' that match the pattern '**/bandit.out'
[Bandit] Traversing of symbolic links: enabled
[Bandit] -> found 1 file
[Bandit] Skipping file 'bandit.out' because it's empty
[Bandit] Skipping post processing
[Bandit] No filter has been set, publishing all 0 issues
[Bandit] Repository miner is not configured, skipping repository mining
[Bandit] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes' that match the pattern '**/bandit.out'
[Bandit] Traversing of symbolic links: enabled
[Bandit] -> found 1 file
[Bandit] Skipping file 'bandit.out' because it's empty
[Bandit] Skipping post processing
[Bandit] No filter has been set, publishing all 0 issues
[Bandit] Repository miner is not configured, skipping repository mining
[Bandit] Reference build recorder is not configured
[Bandit] No valid reference build found
[Bandit] All reported issues will be considered outstanding
[Bandit] Evaluating quality gates
[Bandit] -> All quality gates have been passed
[Bandit] -> Details for each quality gate:
[Bandit]    - [Total (any severity)]: ≪Correcto≫ - (Actual value: 0, Quality gate: 2,00)
[Bandit]    - [Total (any severity)]: ≪Correcto≫ - (Actual value: 0, Quality gate: 4,00)
[Bandit] Health report is disabled - skipping
[Bandit] Created analysis result for 0 issues (found 0 new issues, fixed 0 issues)
[Bandit] Attaching ResultAction with ID 'pylint' to build 'Unir/JENKINSFILE_agentes #13'.
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] recordIssues
WARNING: Unknown parameter(s) found for class type 'io.jenkins.plugins.analysis.core.util.WarningsQualityGate': healthy
[Flake8] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes' that match the pattern '**/flake8.out'
[Flake8] Traversing of symbolic links: enabled
[Flake8] -> found 1 file
[Flake8] Successfully parsed file C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes\flake8.out
[Flake8] -> found 9 issues (skipped 0 duplicates)
[Flake8] Successfully processed file 'flake8.out'
[Flake8] Post processing issues on 'agent1' with source code encoding 'windows-1252'
[Flake8] Creating SCM blamer to obtain author and commit information for affected files
[Flake8] -> No blamer installed yet. You need to install the 'git-forensics' plugin to enable blaming for Git.
[Flake8] Resolving file names for all issues in workspace 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes'
[Flake8] -> resolved paths in source directory (2 found, 0 not found)
[Flake8] Resolving module names from module definitions (build.xml, pom.xml, or Manifest.mf files)
[Flake8] -> resolved module names for 9 issues
[Flake8] Resolving package names (or namespaces) by parsing the affected files
[Flake8] -> resolved package names of 2 affected files
[Flake8] No filter has been set, publishing all 9 issues
[Flake8] Creating fingerprints for all affected code blocks to track issues over different builds
[Flake8] -> created fingerprints for 9 issues (skipped 0 issues)
[Flake8] Copying affected files to Jenkins' build folder 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes\builds\13\files-with-issues'
[Flake8] -> 2 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[Flake8] Skipping cleaning of source code files in old builds
[Flake8] Repository miner is not configured, skipping repository mining
[Flake8] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes' that match the pattern '**/flake8.out'
[Flake8] Traversing of symbolic links: enabled
[Flake8] -> found 1 file
[Flake8] Successfully parsed file C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes\flake8.out
[Flake8] -> found 9 issues (skipped 0 duplicates)
[Flake8] Successfully processed file 'flake8.out'
```

```
[Flake8] Post processing issues on 'agent1' with source code encoding 'windows-1252'
[Flake8] Creating SCM blamer to obtain author and commit information for affected files
[Flake8] -> No blamer installed yet. You need to install the 'git-forensics' plugin to enable blaming for Git.
[Flake8] Resolving file names for all issues in workspace 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes'
[Flake8] -> resolved paths in source directory (2 found, 0 not found)
[Flake8] Resolving module names from module definitions (build.xml, pom.xml, or Manifest.mf files)
[Flake8] -> resolved module names for 9 issues
[Flake8] Resolving package names (or namespaces) by parsing the affected files
[Flake8] -> resolved package names of 2 affected files
[Flake8] No filter has been set, publishing all 9 issues
[Flake8] Creating fingerprints for all affected code blocks to track issues over different builds
[Flake8] -> created fingerprints for 9 issues (skipped 0 issues)
[Flake8] Copying affected files to Jenkins' build folder 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes\builds\13\files-with-issues'
[Flake8] -> 2 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[Flake8] Skipping cleaning of source code files in old builds
[Flake8] Repository miner is not configured, skipping repository mining
[Flake8] Reference build recorder is not configured
[Flake8] No valid reference build found
[Flake8] All reported issues will be considered outstanding
[Flake8] Evaluating quality gates
[Flake8] -> Some quality gates have been missed: overall result is UNSTABLE
[Flake8] -> Details for each quality gate:
[Flake8]    - [Total (any severity)]: ≪Inestable≫ - (Actual value: 9, Quality gate: 8,00)
[Flake8]    - [Total (any severity)]: ≪Correcto≫ - (Actual value: 9, Quality gate: 10,00)
[Flake8] Health report is disabled - skipping
[Flake8] Created analysis result for 9 issues (found 0 new issues, fixed 0 issues)
[Flake8] Attaching ResultAction with ID 'flake8' to build 'Unir/JENKINSFILE_agentes #13'.
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // parallel
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Coverage)
[Pipeline] node
Running on agent1 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>C:\Users\amaro\AppData\Local\Programs\Python\Python313\Scripts\coverage.exe run --branch --source=app --omit=app\__init__.py,app\api.py -m pytest test\unit
============================= test session starts =============================
platform win32 -- Python 3.13.0, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
configfile: pytest.ini
collected 10 items

test\unit\calc_test.py ........                                          [ 80%]
test\unit\util_test.py ..                                                [100%]

============================== 10 passed in 0.04s ==============================

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>C:\Users\amaro\AppData\Local\Programs\Python\Python313\Scripts\coverage.exe xml
```

```
Wrote XML report to coverage.xml
[Pipeline] catchError
[Pipeline] {
[Pipeline] cobertura
[Cobertura] Skipping Cobertura coverage report as build was not SUCCESS or better ...

[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Performance)
[Pipeline] node
Running on agent3 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>SET FLASK_APP=app\api.py

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>start                                    /B
C:\Users\amaro\AppData\Local\Programs\Python\Python313\python.exe -m flask run --host=0.0.0.0 --port=5000

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>timeout  /t  20  /nobreak    // Espera  10
segundos para asegurarse de que Flask estÃ© listo antes de correr las pruebas
ERROR: Sintaxis no v lida. La opci¢n predeterminada no est  permitida m s
de "1" veces.
Escriba "TIMEOUT /?" para su uso.

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>C:\Users\amaro\Downloads\apache-
jmeter-5.6.3\apache-jmeter-5.6.3\bin\jmeter -n -t test\jmeter\flask.jmx -f -l flask.jtl
Usage: python -m flask run [OPTIONS]
Try 'python -m flask run --help' for help.

Error: Could not import 'api'.
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
Creating summariser <summary>
Created the tree successfully using test\jmeter\flask.jmx
Starting standalone test @ 2025 Jan 20 21:51:25 CET (1737406285121)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary =    160 in 00:00:01 =  163.1/s Avg:    2 Min:    1 Max:   22 Err:    0 (0.00%)
Tidying up ...    @ 2025 Jan 20 21:51:26 CET (1737406286166)
... end of run
[Pipeline] script
[Pipeline] {
[Pipeline] perfReport
Creating parser with percentiles:'0,50,90,95,100,' filterRegex:null
Performance: Recording JMeterCsv reports '**/flask.jtl'
Performance:                JMeterCsv                copying                reports                to                master,                files
'[C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes\flask.jtl]'
Performance: JMeterCsv parsing local reports '[C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes\builds\13\performance-
reports\JMeterCSV\flask.jtl]'
Performance:    Parsing    report    file    'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes\builds\13\performance-
reports\JMeterCSV\flask.jtl' with filterRegex 'null'.
Performance: No threshold configured for making the test unstable
Performance: No threshold configured for making the test failure
```

```
Performance: File flask.jtl reported 0.0% of errors [SUCCESS]. Build status is: UNSTABLE
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] End of Pipeline
Finished: UNSTABLE
```

Lo que sucede en el pipeline:

- **Etapas en paralelo**: En el stage "Static and Security Analysis", hay dos etapas ("Static" y "Security") que se ejecutan en paralelo. Esto significa que Jenkins intentará ejecutar ambas etapas de manera simultánea, lo que se ve beneficiado por los 3 ejecutores de los agentes correspondientes (en este caso, "agent1" para la etapa de Static y "agent2" para la etapa de Security).
    - Como tienes 3 ejecutores en cada agente, si tienes varias tareas en un mismo agente, se ejecutarán al mismo tiempo, siempre y cuando no haya otras restricciones (por ejemplo, otras tareas en espera de los ejecutores disponibles).
- **Uso de recursos en paralelo**: Al tener 3 ejecutores en cada agente, si tienes varias etapas en un mismo agente (como ocurre con "agent1"), Jenkins podrá ejecutar varias de esas tareas en paralelo, sin tener que esperar que se liberen otros ejecutores. Esto aumenta la eficiencia y disminuye el tiempo total de ejecución del pipeline.

Impacto del uso de 3 ejecutores:

- **Eficiencia**: Tener múltiples ejecutores permite que Jenkins ejecute varios trabajos a la vez, reduciendo el tiempo total de ejecución del pipeline, especialmente si tienes muchas etapas que requieren un agente.
- **Posible sobrecarga**: Si el número de ejecutores por agente es mayor que la cantidad de trabajos que el agente puede manejar efectivamente, podrías estar desperdiciando recursos. Sin embargo, tener múltiples ejecutores es útil si tus tareas pueden aprovechar la concurrencia y si tu agente tiene suficientes recursos de hardware (CPU, memoria, etc.) para soportar múltiples ejecuciones al mismo tiempo.

Resumen:

Con **3 ejecutores por agente**, Jenkins puede ejecutar simultáneamente hasta 3 trabajos diferentes en el mismo agente. Esto permite un mejor uso de los recursos y acelera la ejecución de tareas concurrentes. En tu pipeline, cuando se ejecutan tareas paralelas o en la misma máquina, estas podrán

ejecutarse simultáneamente si están en diferentes ejecutores. Esto mejora la eficiencia general del pipeline y reduce los tiempos de espera.

- Log y explicación sobre lo que ocurre cuando el número de executors se reduce a 1, teniendo 4-5-6 etapas ejecutándose simultáneamente en 2-3 agentes.

```
Lanzada por el usuario Giovanna leon
[Pipeline] Start of Pipeline
[Pipeline] stage
[Pipeline] { (Get Code)
[Pipeline] node
Running on agent1 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Fetching changes from the remote Git repository
 > git.exe  rev-parse  --resolve-git-dir  C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes\.git  # timeout=10
 > git.exe config remote.origin.url https://github.com/GiovannaLeon/helloworld.git # timeout=10
Fetching upstream changes from https://github.com/GiovannaLeon/helloworld.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.47.1.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/GiovannaLeon/helloworld.git +refs/heads/*:refs/remotes/origin/* # timeout=10
Checking out Revision eb8d8309fcd89b880e7bc330d180dfeceaf53b9a (refs/remotes/origin/master)
Commit message: "Add files via upload"
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe branch -D master # timeout=10
 > git.exe checkout -b master eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
 > git.exe rev-list --no-walk eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>dir
 El volumen de la unidad C es Windows
 El n£mero de serie del volumen es: 6475-04AA

 Directorio de C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes

20/01/2025  23:04    <DIR>          .
20/01/2025  01:38    <DIR>          ..
20/01/2025  23:04            53.248 .coverage
20/01/2025  01:38                40 .gitignore
20/01/2025  01:38    <DIR>          .pytest_cache
20/01/2025  01:38    <DIR>          app
20/01/2025  23:04             2.273 coverage.xml
20/01/2025  23:04               500 flake8.out
20/01/2025  01:38    <DIR>          jenkinsFile_1
20/01/2025  01:38    <DIR>          jenkinsFile_2
20/01/2025  01:38    <DIR>          JenkinsFile_3
20/01/2025  01:38    <DIR>          jenkinsfile_4
```

```
20/01/2025  01:38               175 pytest.ini
20/01/2025  01:38               418 README.md
20/01/2025  23:04             1.401 result-unit.xml
20/01/2025  01:38    <DIR>          test
               7 archivos         58.055 bytes
               9 dirs  564.906.209.280 bytes libres
[Pipeline] echo
Workspace: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Unit)
[Pipeline] node
Running on agent1 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] catchError
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>SET
PYTHONPATH=C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>C:\Users\amaro\AppData\Local\Programs\
Python\Python313\python.exe -m pytest --junitxml=result-unit.xml test\unit
============================= test session starts =============================
platform win32 -- Python 3.13.0, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
configfile: pytest.ini
collected 10 items

test\unit\calc_test.py .....                                [ 80%]
test\unit\util_test.py ..                                   [100%]

- generated xml file: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes\result-unit.xml -
============================= 10 passed in 0.10s ==============================
[Pipeline] sleep
Sleeping for 5 Seg
[Pipeline] junit
Grabando resultados de tests
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Static and Security Analysis)
[Pipeline] parallel
[Pipeline] { (Branch: Static)
[Pipeline] { (Branch: Security)
[Pipeline] stage
[Pipeline] { (Static)
[Pipeline] stage
[Pipeline] { (Security)
[Pipeline] node
Running on agent1 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] node
```

```
Running on agent2 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes
[Pipeline] {
[Pipeline] {
[Pipeline] bat
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>C:\Users\amaro\AppData\Local\Programs\
Python\Python313\Scripts\flake8.exe --exit-zero --format=pylint --exit-zero app  1>flake8.out

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes>C:\Users\amaro\AppData\Local\Programs\
Python\Python313\Scripts\bandit.exe --exit-zero -r . -f custom -o bandit.out --msg-template "{abspath}:{line}: [{test_id}] {msg}"
[main]      INFO    profile include tests: None
[main]      INFO    profile exclude tests: None
[main]      INFO    cli include tests: None
[main]      INFO    cli exclude tests: None
[main]      INFO    running on Python 3.13.0
[custom]    INFO    Result written to file: bandit.out
[Pipeline] catchError
[Pipeline] {
[Pipeline] recordIssues
WARNING: Unknown parameter(s) found for class type 'io.jenkins.plugins.analysis.core.util.WarningsQualityGate': healthy
[Bandit] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes' that match
the pattern '**/bandit.out'
[Bandit] Traversing of symbolic links: enabled
[Bandit] -> found 1 file
[Bandit] Skipping file 'bandit.out' because it's empty
[Bandit] Skipping post processing
[Bandit] No filter has been set, publishing all 0 issues
[Bandit] Repository miner is not configured, skipping repository mining
[Bandit] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes' that match
the pattern '**/bandit.out'
[Bandit] Traversing of symbolic links: enabled
[Bandit] -> found 1 file
[Bandit] Skipping file 'bandit.out' because it's empty
[Bandit] Skipping post processing
[Bandit] No filter has been set, publishing all 0 issues
[Bandit] Repository miner is not configured, skipping repository mining
[Bandit] Reference build recorder is not configured
[Bandit] No valid reference build found
[Bandit] All reported issues will be considered outstanding
[Bandit] Evaluating quality gates
[Bandit] -> All quality gates have been passed
[Bandit] -> Details for each quality gate:
[Bandit]    - [Total (any severity)]: ≪Correcto≫ - (Actual value: 0, Quality gate: 2,00)
[Bandit]    - [Total (any severity)]: ≪Correcto≫ - (Actual value: 0, Quality gate: 4,00)
[Bandit] Health report is disabled - skipping
[Bandit] Created analysis result for 0 issues (found 0 new issues, fixed 0 issues)
```

```
[Bandit] Attaching ResultAction with ID 'pylint' to build 'Unir/JENKINSFILE_agentes #15'.
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] recordIssues
WARNING: Unknown parameter(s) found for class type 'io.jenkins.plugins.analysis.core.util.WarningsQualityGate': healthy
[Flake8] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes' that match the pattern '**/flake8.out'
[Flake8] Traversing of symbolic links: enabled
[Flake8] -> found 1 file
[Flake8] Successfully parsed file C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes\flake8.out
[Flake8] -> found 9 issues (skipped 0 duplicates)
[Flake8] Successfully processed file 'flake8.out'
[Flake8] Post processing issues on 'agent1' with source code encoding 'windows-1252'
[Flake8] Creating SCM blamer to obtain author and commit information for affected files
[Flake8] -> No blamer installed yet. You need to install the 'git-forensics' plugin to enable blaming for Git.
[Flake8] Resolving file names for all issues in workspace 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes'
[Flake8] -> resolved paths in source directory (2 found, 0 not found)
[Flake8] Resolving module names from module definitions (build.xml, pom.xml, or Manifest.mf files)
[Flake8] -> resolved module names for 9 issues
[Flake8] Resolving package names (or namespaces) by parsing the affected files
[Flake8] -> resolved package names of 2 affected files
[Flake8] No filter has been set, publishing all 9 issues
[Flake8] Creating fingerprints for all affected code blocks to track issues over different builds
[Flake8] -> created fingerprints for 9 issues (skipped 0 issues)
[Flake8] Copying affected files to Jenkins' build folder 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes\builds\15\files-with-issues'
[Flake8] -> 2 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[Flake8] Skipping cleaning of source code files in old builds
[Flake8] Repository miner is not configured, skipping repository mining
[Flake8] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes' that match the pattern '**/flake8.out'
[Flake8] Traversing of symbolic links: enabled
[Flake8] -> found 1 file
[Flake8] Successfully parsed file C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes\flake8.out
[Flake8] -> found 9 issues (skipped 0 duplicates)
[Flake8] Successfully processed file 'flake8.out'
[Flake8] Post processing issues on 'agent1' with source code encoding 'windows-1252'
[Flake8] Creating SCM blamer to obtain author and commit information for affected files
[Flake8] -> No blamer installed yet. You need to install the 'git-forensics' plugin to enable blaming for Git.
[Flake8] Resolving file names for all issues in workspace 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes'
[Flake8] -> resolved paths in source directory (2 found, 0 not found)
[Flake8] Resolving module names from module definitions (build.xml, pom.xml, or Manifest.mf files)
[Flake8] -> resolved module names for 9 issues
[Flake8] Resolving package names (or namespaces) by parsing the affected files
[Flake8] -> resolved package names of 2 affected files
[Flake8] No filter has been set, publishing all 9 issues
[Flake8] Creating fingerprints for all affected code blocks to track issues over different builds
[Flake8] -> created fingerprints for 9 issues (skipped 0 issues)
[Flake8] Copying affected files to Jenkins' build folder 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes\builds\15\files-with-issues'
[Flake8] -> 2 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[Flake8] Skipping cleaning of source code files in old builds
[Flake8] Repository miner is not configured, skipping repository mining
[Flake8] Reference build recorder is not configured
[Flake8] No valid reference build found
[Flake8] All reported issues will be considered outstanding
[Flake8] Evaluating quality gates
[Flake8] -> Some quality gates have been missed: overall result is UNSTABLE
[Flake8] -> Details for each quality gate:
[Flake8]    - [Total (any severity)]: ≪Inestable≫ - (Actual value: 9, Quality gate: 8,00)
[Flake8]    - [Total (any severity)]: ≪Correcto≫ - (Actual value: 9, Quality gate: 10,00)
[Flake8] Health report is disabled - skipping
[Flake8] Created analysis result for 9 issues (found 0 new issues, fixed 0 issues)
[Flake8] Attaching ResultAction with ID 'flake8' to build 'Unir/JENKINSFILE_agentes #15'.
[Checks API] No suitable checks publisher found.
```

```
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // parallel
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Coverage)
[Pipeline] node
Running on agent1 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>C:\Users\amaro\AppData\Local\Programs\
Python\Python313\Scripts\coverage.exe run --branch --source=app --omit=app\__init__.py,app\api.py -m pytest test\unit
============================= test session starts =============================
platform win32 -- Python 3.13.0, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes
configfile: pytest.ini
collected 10 items

test\unit\calc_test.py .....                                [ 80%]
test\unit\util_test.py ..                                   [100%]

============================== 10 passed in 0.03s ==============================

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes>C:\Users\amaro\AppData\Local\Programs\
Python\Python313\Scripts\coverage.exe xml
Wrote XML report to coverage.xml
[Pipeline] catchError
[Pipeline] {
[Pipeline] cobertura
[Cobertura] Skipping Cobertura coverage report as build was not SUCCESS or better ...

[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Performance)
[Pipeline] node
Running on agent3 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes
```

```
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>SET FLASK_APP=app\api.py

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>start /B
C:\Users\amaro\AppData\Local\Programs\Python\Python313\python.exe -m flask run --host=0.0.0.0 --port=5000

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>timeout /t 20 /nobreak // Espera 10
segundos para asegurarse de que Flask estÃ© listo antes de correr las pruebas
ERROR: Sintaxis no v lida. La opci¢n predeterminada no est permitida m s
de "1" veces.
Escriba "TIMEOUT /?" para su uso.

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes>C:\Users\amaro\Downloads\apache-
jmeter-5.6.3\apache-jmeter-5.6.3\bin\jmeter -n -t test\jmeter\flask.jmx -f -l flask.jtl
Usage: python -m flask run [OPTIONS]
Try 'python -m flask run --help' for help.

Error: Could not import 'api'.
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
Creating summariser <summary>
Created the tree successfully using test\jmeter\flask.jmx
Starting standalone test @ 2025 Jan 20 23:16:52 CET (1737411412267)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary =    160 in 00:00:01 =  161.9/s Avg:    2 Min:    1 Max:    23 Err:    0 (0.00%)
Tidying up ...    @ 2025 Jan 20 23:16:53 CET (1737411413322)
... end of run
[Pipeline] script
[Pipeline] {
[Pipeline] perfReport
Creating parser with percentiles:'0,50,90,95,100,' filterRegex:null
Performance: Recording JMeterCsv reports '**/flask.jtl'
Performance: JMeterCsv copying reports to master, files
'[C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes\flask.jtl]'
Performance: JMeterCsv parsing local reports '[C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes\builds\15\performance-
reports\JMeterCSV\flask.jtl]'
Performance: Parsing report file 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes\builds\15\performance-
reports\JMeterCSV\flask.jtl' with filterRegex 'null'.
Performance: No threshold configured for making the test unstable
Performance: No threshold configured for making the test failure
Performance: File flask.jtl reported 0.0% of errors [SUCCESS]. Build status is: UNSTABLE
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] End of Pipeline
Finished: UNSTABLE
```

Reducir el número de **executors** a **1** por agente disminuye la capacidad de ejecución simultánea de Jenkins. En un pipeline con múltiples etapas, especialmente cuando tienes muchas etapas que necesitan ejecutarse en paralelo en **varios agentes**, reducir los ejecutores provoca **cuellos de botella** y **tiempos de espera**, lo que aumenta el tiempo total de ejecución. Las etapas se ejecutarán de manera **secuencial en cada agente**, y las tareas tendrán que esperar a que se liberen los ejecutores disponibles, lo que afecta la eficiencia del pipeline. Esto podría ser una limitación si tu infraestructura o pipeline tiene una gran cantidad de tareas concurrentes o si las tareas son pesadas

- Explicación de la separación realizada y fundamentación de la misma.

1. Separación de Agentes:

La decisión de separar las etapas en distintos agentes tiene varias motivaciones estratégicas y técnicas:

A. Optimización de recursos:

- **agent1** está asignado a las etapas de **'Get Code'**, **'Unit'**, **'Static'** y **'Coverage'**. Este agente parece estar optimizado para tareas de desarrollo, como la compilación, las pruebas unitarias, el análisis estático y la cobertura de pruebas.
    - Al agrupar estas tareas en un solo agente, se garantiza que el entorno de desarrollo, las dependencias y las herramientas específicas estén preinstaladas y configuradas correctamente en un único lugar, lo que mejora la coherencia de los resultados y la eficiencia del pipeline.
- **agent2** está dedicado a la etapa de **'Security'** (análisis de seguridad con Bandit). Se utiliza un agente diferente para realizar el análisis de seguridad porque, generalmente, este tipo de herramientas puede requerir una configuración específica o permisos adicionales (por ejemplo, configuraciones de seguridad, entornos de red aislados, etc.).
    - Utilizar un agente separado para las pruebas de seguridad reduce el riesgo de interferencias con las tareas de desarrollo o de ejecución de código en el mismo entorno, y también permite configurar este agente con políticas de seguridad adicionales sin afectar el flujo principal de integración.
- **agent3** está asignado a la etapa de **'Performance'** (pruebas de rendimiento con JMeter y ejecución de Flask). Este agente está dedicado a tareas de rendimiento, que generalmente requieren recursos adicionales y configuraciones específicas, como configuraciones de red o servidores.
    - Al usar un agente especializado para pruebas de rendimiento, se evita que estas pruebas interfieran con las etapas de desarrollo y pruebas unitarias que podrían tener un impacto en el rendimiento del servidor.

B. Aislamiento y Especialización:

Separar las tareas en diferentes agentes permite que cada uno esté especializado para un conjunto específico de tareas, lo que facilita la **gestión de dependencias** y evita problemas de **interferencia** entre las distintas fases del pipeline.

- **Entornos de ejecución diferenciados:** Algunas etapas, como las de seguridad (con Bandit) y de pruebas de rendimiento, pueden requerir un acceso más restringido o una configuración

diferente, como privilegios elevados para ejecutar herramientas específicas, o la necesidad de acceso a bases de datos o servidores de prueba.

- **Facilidad de mantenimiento:** Si se encuentra un error en una de las etapas (por ejemplo, en las pruebas de seguridad), el cambio puede realizarse de manera aislada en el agente correspondiente sin afectar a los demás agentes, lo que facilita la **gestión de problemas**.

2. Creación de Agentes: Métodos Empleados

Existen varias formas de crear y gestionar agentes en Jenkins, los cuales pueden ser **agentes físicos** o **virtuales**. El método empleado para la creación de estos agentes no se especifica explícitamente en el pipeline, pero se puede inferir que probablemente se están utilizando **agentes remotos gestionados por SSH** o **agentes basados en Java** (usualmente Jenkins master/slave).

A. Agentes gestionados por SSH:

Una de las formas más comunes de crear agentes en Jenkins es mediante **SSH**. El master de Jenkins se comunica con un nodo esclavo a través de un servidor SSH para ejecutar tareas.

- **Configuración**: Para cada agente (agent1, agent2, agent3), se debe configurar una conexión SSH que permita a Jenkins ejecutar comandos de manera remota. Esto requiere que los agentes estén configurados con una clave SSH compartida y que Jenkins tenga los permisos adecuados.
- **Ventaja**: Esta configuración permite que Jenkins ejecute tareas de manera distribuida y remota, lo que facilita la **escalabilidad** y la **flexibilidad**, ya que se pueden agregar o quitar agentes fácilmente.
- **Seguridad**: La comunicación a través de SSH es **segura** si se utiliza cifrado adecuado, pero también puede ser **más compleja** de administrar, especialmente cuando se tienen múltiples agentes.

B. Agentes basados en Java (Jenkins slave):

Otra opción es usar agentes que se conectan al master de Jenkins utilizando el protocolo Java Web Start o mediante un cliente Java.

- **Configuración**: Los agentes basados en Java se configuran descargando un archivo JAR desde el servidor maestro de Jenkins y ejecutándolo en el nodo esclavo.
- **Ventaja**: Esta configuración puede ser más sencilla para administrar si la infraestructura de Jenkins es completamente interna, ya que no requiere configuraciones adicionales de red o SSH.
- **Seguridad**: La **comunicación entre master y slave** en Jenkins es cifrada de forma predeterminada, lo que garantiza **seguridad**. Sin embargo, los agentes que se conectan con

el protocolo Java deben ser cuidadosamente monitoreados, ya que pueden ser más vulnerables a **fallos de seguridad** si no se mantienen actualizados.

3. Conclusiones en términos de seguridad y eficiencia:

A. Seguridad:

- **Aislamiento de tareas críticas**: Al utilizar **agentes diferentes** para tareas como seguridad (Bandit) y rendimiento (JMeter), se garantiza que cualquier vulnerabilidad o fallo en una de las etapas no afecte a las demás. Esto es especialmente importante si las pruebas de seguridad implican escaneos de código fuente o análisis de vulnerabilidades.
- **Control de acceso**: Puedes aplicar políticas de seguridad **más estrictas** a agentes que ejecutan pruebas de seguridad o pruebas de rendimiento, como restringir el acceso a recursos sensibles, controlar qué herramientas están disponibles, o asegurar que solo los usuarios autorizados puedan acceder a los agentes.
- **Reducción de riesgos**: Separar las tareas también reduce el riesgo de que un error en una etapa (por ejemplo, una vulnerabilidad en las pruebas de rendimiento) afecte a la estabilidad de las demás etapas del pipeline.

B. Eficiencia:

- **Paralelismo eficiente**: La configuración de **etapas paralelas** (como Static y Security) en diferentes agentes permite que el pipeline ejecute múltiples tareas al mismo tiempo, lo que acelera significativamente el proceso global.
- **Especialización**: Asignar agentes específicos para cada tipo de tarea (desarrollo, seguridad, rendimiento) mejora la eficiencia en cada una de las etapas, ya que cada agente puede estar optimizado para la tarea que realiza. Por ejemplo, el agente de rendimiento (agent3) puede tener más recursos disponibles para ejecutar pruebas de carga sin que afecte las demás tareas.

C. Escalabilidad:

- La infraestructura distribuida, donde diferentes agentes están configurados para diferentes tareas, facilita la **escalabilidad**. Si necesitas más capacidad de procesamiento para una etapa (por ejemplo, más agentes para pruebas de rendimiento), puedes agregar más **nodos de Jenkins** sin interrumpir las tareas que ya están en curso.

Conclusión Final:

Esta distribución de tareas en Jenkins, utilizando **agentes separados**, no solo mejora la **seguridad** al aislar tareas críticas y sensibles, sino que también optimiza la **eficiencia** y **escalabilidad** del pipeline. Utilizar **SSH** o **Java** para gestionar estos agentes depende de la infraestructura y necesidades específicas, pero ambas opciones pueden ser efectivas si se configuran y mantienen adecuadamente.

Reto 3 – Mejora de la cobertura

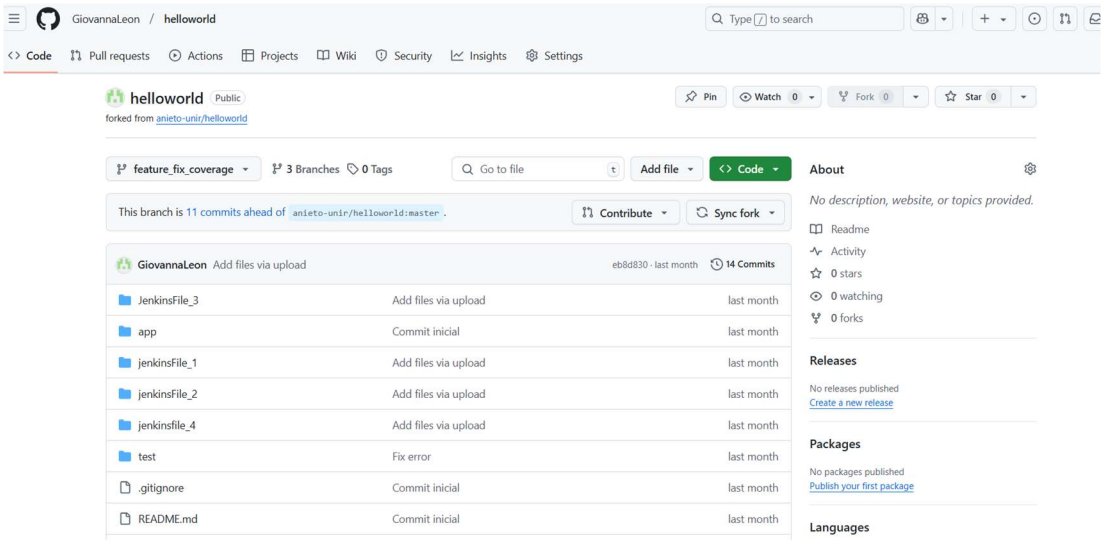
En este reto se solicitan 3 entregables:

- URL al repositorio creado por el alumno, a partir del código fuente base de este CP1, que albergue tanto el código fuente como el nuevo Jenkinsfile.

https://github.com/GiovannaLeon/helloworld.git

El repositorio será el mismo, por lo que no es necesario indicar ninguna otra URL.

Hay que tener en cuenta que ahora tendremos una nueva rama en el repositorio "feature_fix_coverage".



- Log de ejecución del pipeline, sobre rama master, donde se muestren también los datos de cobertura de código por líneas y ramas (que no será cobertura completa).

Lanzada por el usuario Giovanna leon
[Pipeline] Start of Pipeline
[Pipeline] stage
[Pipeline] { (Get Code)
[Pipeline] node
Running on agent1 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] {
[Pipeline] bat

```
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Fetching changes from the remote Git repository
Checking out Revision eb8d8309fcd89b880e7bc330d180dfeceaf53b9a (refs/remotes/origin/master)
 >                                  git.exe                                  rev-parse                                  --resolve-git-dir
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master\.git # timeout=10
 > git.exe config remote.origin.url https://github.com/GiovannaLeon/helloworld.git # timeout=10
Fetching upstream changes from https://github.com/GiovannaLeon/helloworld.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.47.1.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/GiovannaLeon/helloworld.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
Commit message: "Add files via upload"
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe branch -D master # timeout=10
 > git.exe checkout -b master eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
 > git.exe rev-list --no-walk eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>git checkout master
Already on 'master'
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>dir
 El volumen de la unidad C es Windows
 El n£mero de serie del volumen es: 6475-04AA

 Directorio de C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master

23/01/2025  17:02    <DIR>          .
23/01/2025  16:51    <DIR>          ..
23/01/2025  17:02            53.248 .coverage
23/01/2025  16:51                40 .gitignore
23/01/2025  16:51    <DIR>          .pytest_cache
23/01/2025  16:51    <DIR>          app
23/01/2025  17:02             2.280 coverage.xml
23/01/2025  17:02               500 flake8.out
23/01/2025  16:51    <DIR>          jenkinsFile_1
23/01/2025  16:51    <DIR>          jenkinsFile_2
23/01/2025  16:51    <DIR>          JenkinsFile_3
23/01/2025  16:51    <DIR>          jenkinsfile_4
23/01/2025  16:51               175 pytest.ini
23/01/2025  16:51               418 README.md
23/01/2025  17:02             1.401 result-unit.xml
23/01/2025  16:51    <DIR>          test
               7 archivos         58.062 bytes
               9 dirs  562.803.441.664 bytes libres
[Pipeline] echo
Workspace: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Unit)
[Pipeline] node
Running on agent1 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] {
[Pipeline] bat
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] catchError
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>SET
PYTHONPATH=C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>C:\Users\amaro\AppData\Local\Pr
ograms\Python\Python313\python.exe -m pytest --junitxml=result-unit.xml test\unit
============================= test session starts =============================
platform win32 -- Python 3.13.0, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
configfile: pytest.ini
collected 10 items

test\unit\calc_test.py .....                                [ 80%]
test\unit\util_test.py ..                                   [100%]

- generated xml file: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master\result-unit.xml
-
============================== 10 passed in 0.11s ==============================
[Pipeline] sleep
Sleeping for 5 Seg
[Pipeline] junit
Grabando resultados de tests
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Static and Security Analysis)
[Pipeline] parallel
[Pipeline] { (Branch: Static)
[Pipeline] { (Branch: Security)
[Pipeline] stage
[Pipeline] { (Static)
[Pipeline] stage
[Pipeline] { (Security)
[Pipeline] node
Running on agent1 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] node
Running on agent2 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] {
[Pipeline] {
[Pipeline] bat
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_master>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>hostname
Amaro
[Pipeline] bat
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_master>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_master>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_master
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>C:\Users\amaro\AppData\Local\Pr
ograms\Python\Python313\Scripts\flake8.exe --exit-zero --format=pylint --exit-zero app  1>flake8.out

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_master>C:\Users\amaro\AppData\Local\Pr
ograms\Python\Python313\Scripts\bandit.exe --exit-zero -r . -f custom -o bandit.out --msg-template "{abspath}:{line}: [{test_id}] {msg}"
[main]      INFO        profile include tests: None
[main]      INFO        profile exclude tests: None
[main]      INFO        cli include tests: None
[main]      INFO        cli exclude tests: None
[main]      INFO        running on Python 3.13.0
[custom]    INFO        Result written to file: bandit.out
[Pipeline] catchError
[Pipeline] {
[Pipeline] recordIssues
WARNING: Unknown parameter(s) found for class type 'io.jenkins.plugins.analysis.core.util.WarningsQualityGate': healthy
[Bandit] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_master'
that match the pattern '**/bandit.out'
[Bandit] Traversing of symbolic links: enabled
[Bandit] -> found 1 file
[Bandit] Skipping file 'bandit.out' because it's empty
[Bandit] Skipping post processing
[Bandit] No filter has been set, publishing all 0 issues
[Bandit] Repository miner is not configured, skipping repository mining
[Bandit] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_master'
that match the pattern '**/bandit.out'
[Bandit] Traversing of symbolic links: enabled
[Bandit] -> found 1 file
[Bandit] Skipping file 'bandit.out' because it's empty
[Bandit] Skipping post processing
[Bandit] No filter has been set, publishing all 0 issues
[Bandit] Repository miner is not configured, skipping repository mining
[Bandit] Reference build recorder is not configured
[Bandit] No valid reference build found
[Bandit] All reported issues will be considered outstanding
[Bandit] Evaluating quality gates
[Bandit] -> All quality gates have been passed
[Bandit] -> Details for each quality gate:
[Bandit]    - [Total (any severity)]: ≪Correcto≫ - (Actual value: 0, Quality gate: 2,00)
[Bandit]    - [Total (any severity)]: ≪Correcto≫ - (Actual value: 0, Quality gate: 4,00)
[Bandit] Health report is disabled - skipping
[Bandit] Created analysis result for 0 issues (found 0 new issues, fixed 0 issues)
[Bandit] Attaching ResultAction with ID 'pylint' to build 'Unir/JENKINSFILE_agentes_master #8'.
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] recordIssues
WARNING: Unknown parameter(s) found for class type 'io.jenkins.plugins.analysis.core.util.WarningsQualityGate': healthy
[Flake8] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master'
that match the pattern '**/flake8.out'
[Flake8] Traversing of symbolic links: enabled
[Flake8] -> found 1 file
[Flake8] Successfully parsed file
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master\flake8.out
[Flake8] -> found 9 issues (skipped 0 duplicates)
[Flake8] Successfully processed file 'flake8.out'
```

```
[Pipeline] // stage
[Pipeline] }
[Flake8] Post processing issues on 'agent1' with source code encoding 'windows-1252'
[Flake8] Creating SCM blamer to obtain author and commit information for affected files
[Flake8] -> No blamer installed yet. You need to install the 'git-forensics' plugin to enable blaming for Git.
[Flake8] Resolving file names for all issues in workspace 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master'
[Flake8] -> resolved paths in source directory (2 found, 0 not found)
[Flake8] Resolving module names from module definitions (build.xml, pom.xml, or Manifest.mf files)
[Flake8] -> resolved module names for 9 issues
[Flake8] Resolving package names (or namespaces) by parsing the affected files
[Flake8] -> resolved package names of 2 affected files
[Flake8] No filter has been set, publishing all 9 issues
[Flake8] Creating fingerprints for all affected code blocks to track issues over different builds
[Flake8] -> created fingerprints for 9 issues (skipped 0 issues)
[Flake8] Copying affected files to Jenkins' build folder 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes_master\builds\8\files-with-issues'
[Flake8] -> 2 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[Flake8] Skipping cleaning of source code files in old builds
[Flake8] Repository miner is not configured, skipping repository mining
[Flake8] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master' that match the pattern '**/flake8.out'
[Flake8] Traversing of symbolic links: enabled
[Flake8] -> found 1 file
[Flake8] Successfully parsed file C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master\flake8.out
[Flake8] -> found 9 issues (skipped 0 duplicates)
[Flake8] Successfully processed file 'flake8.out'
[Flake8] Post processing issues on 'agent1' with source code encoding 'windows-1252'
[Flake8] Creating SCM blamer to obtain author and commit information for affected files
[Flake8] -> No blamer installed yet. You need to install the 'git-forensics' plugin to enable blaming for Git.
[Flake8] Resolving file names for all issues in workspace 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master'
[Flake8] -> resolved paths in source directory (2 found, 0 not found)
[Flake8] Resolving module names from module definitions (build.xml, pom.xml, or Manifest.mf files)
[Flake8] -> resolved module names for 9 issues
[Flake8] Resolving package names (or namespaces) by parsing the affected files
[Flake8] -> resolved package names of 2 affected files
[Flake8] No filter has been set, publishing all 9 issues
[Flake8] Creating fingerprints for all affected code blocks to track issues over different builds
[Flake8] -> created fingerprints for 9 issues (skipped 0 issues)
[Flake8] Copying affected files to Jenkins' build folder 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes_master\builds\8\files-with-issues'
[Flake8] -> 2 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[Flake8] Skipping cleaning of source code files in old builds
[Flake8] Repository miner is not configured, skipping repository mining
[Flake8] Reference build recorder is not configured
[Flake8] No valid reference build found
[Flake8] All reported issues will be considered outstanding
[Flake8] Evaluating quality gates
[Flake8] -> Some quality gates have been missed: overall result is UNSTABLE
[Flake8] -> Details for each quality gate:
[Flake8]    - [Total (any severity)]: ≪Inestable≫ - (Actual value: 9, Quality gate: 8,00)
[Flake8]    - [Total (any severity)]: ≪Correcto≫ - (Actual value: 9, Quality gate: 10,00)
[Flake8] Health report is disabled - skipping
[Flake8] Created analysis result for 9 issues (found 0 new issues, fixed 0 issues)
[Flake8] Attaching ResultAction with ID 'flake8' to build 'Unir/JENKINSFILE_agentes_master #8'.
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // parallel
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Coverage)
[Pipeline] node
Running on agent1 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] {
[Pipeline] bat
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>C:\Users\amaro\AppData\Local\Pr
ograms\Python\Python313\Scripts\coverage.exe run --branch --source=app --omit=app\__init__.py,app\api.py -m pytest test\unit
============================= test session starts =============================
platform win32 -- Python 3.13.0, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master
configfile: pytest.ini
collected 10 items

test\unit\calc_test.py .....                                [ 80%]
test\unit\util_test.py ..                                   [100%]

============================== 10 passed in 0.05s ==============================

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_master>C:\Users\amaro\AppData\Local\Pr
ograms\Python\Python313\Scripts\coverage.exe xml
Wrote XML report to coverage.xml
[Pipeline] catchError
[Pipeline] {
[Pipeline] cobertura
[Cobertura] Skipping Cobertura coverage report as build was not SUCCESS or better ...

[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Performance)
[Pipeline] node
Running on agent3 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_master>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_master>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_master>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_master
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_master
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_master>SET FLASK_APP=app\api.py

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_master>start                 /B
C:\Users\amaro\AppData\Local\Programs\Python\Python313\python.exe -m flask run --host=0.0.0.0 --port=5000

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_master>timeout /t 20 /nobreak  // Espera 10
segundos para asegurarse de que Flask estÃ© listo antes de correr las pruebas
ERROR: Sintaxis no v lida. La opci¢n predeterminada no est  permitida m s
de "1" veces.
Escriba "TIMEOUT /?" para su uso.

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_master>C:\Users\amaro\Downloads\apach
e-jmeter-5.6.3\apache-jmeter-5.6.3\bin\jmeter -n -t test\jmeter\flask.jmx -f -l flask.jtl
```

```
Usage: python -m flask run [OPTIONS]
Try 'python -m flask run --help' for help.

Error: Could not import 'api'.
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
Creating summariser <summary>
Created the tree successfully using test\jmeter\flask.jmx
Starting standalone test @ 2025 Jan 23 17:04:18 CET (1737648258147)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary =    200 in 00:00:01 =  180.3/s Avg:    9 Min:    3 Max:   35 Err:    0 (0.00%)
Tidying up ...    @ 2025 Jan 23 17:04:19 CET (1737648259378)
... end of run
[Pipeline] script
[Pipeline] {
[Pipeline] perfReport
Creating parser with percentiles:'0,50,90,95,100,' filterRegex:null
Performance: Recording JMeterCsv reports '**/flask.jtl'
Performance: JMeterCsv copying reports to master, files '[C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_master\flask.jtl]'
Performance: JMeterCsv parsing local reports '[C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes_master\builds\8\performance-reports\JMeterCSV\flask.jtl]'
Performance: Parsing report file 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes_master\builds\8\performance-reports\JMeterCSV\flask.jtl' with filterRegex 'null'.
Performance: No threshold configured for making the test unstable
Performance: No threshold configured for making the test failure
Performance: File flask.jtl reported 0.0% of errors [SUCCESS]. Build status is: UNSTABLE
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] End of Pipeline
Finished: UNSTABLE
```

- Log de ejecución del pipeline, sobre la nueva rama “feature_fix_coverage”, donde se muestren también los datos de cobertura de código por líneas y ramas, ahora ya con un 100% de cobertura.

```
Lanzada por el usuario Giovanna leon

[Pipeline] Start of Pipeline
[Pipeline] stage
[Pipeline] { (Get Code)
[Pipeline] node
Running on agent1
 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>hostname
```

```
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Fetching changes from the remote Git repository
Checking out Revision eb8d8309fcd89b880e7bc330d180dfeceaf53b9a (refs/remotes/origin/feature_fix_coverage)
 > git.exe rev-parse --resolve-git-dir
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage\.git # timeout=10
 > git.exe config remote.origin.url https://github.com/GiovannaLeon/helloworld.git # timeout=10
Fetching upstream changes from https://github.com/GiovannaLeon/helloworld.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.47.1.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/GiovannaLeon/helloworld.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/feature_fix_coverage^{commit}" # timeout=10
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
Commit message: "Add files via upload"
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe branch -D feature_fix_coverage # timeout=10
 > git.exe checkout -b feature_fix_coverage eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
 > git.exe rev-list --no-walk eb8d8309fcd89b880e7bc330d180dfeceaf53b9a # timeout=10
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>dir
 El volumen de la unidad C es Windows
 El n£mero de serie del volumen es: 6475-04AA

 Directorio de C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage

21/01/2025  00:44    <DIR>          .
23/01/2025  16:51    <DIR>          ..
21/01/2025  00:44            53.248 .coverage
20/01/2025  23:58                40 .gitignore
20/01/2025  23:58    <DIR>          .pytest_cache
20/01/2025  23:58    <DIR>          app
21/01/2025  00:44             2.294 coverage.xml
21/01/2025  00:44               500 flake8.out
20/01/2025  23:58    <DIR>          jenkinsFile_1
20/01/2025  23:58    <DIR>          jenkinsFile_2
20/01/2025  23:58    <DIR>          JenkinsFile_3
20/01/2025  23:58    <DIR>          jenkinsfile_4
20/01/2025  23:58               175 pytest.ini
20/01/2025  23:58               418 README.md
21/01/2025  00:44             1.401 result-unit.xml
20/01/2025  23:58    <DIR>          test
               7 archivos         58.076 bytes
               9 dirs  562.795.507.712 bytes libres
[Pipeline] echo
Workspace: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
```

```
[Pipeline] stage
[Pipeline] { (Unit)
[Pipeline] node
Running on agent1
 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] catchError
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>SET
PYTHONPATH=C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>C:\Users\amaro\App
Data\Local\Programs\Python\Python313\python.exe -m pytest --junitxml=result-unit.xml test\unit
============================= test session starts =============================
platform win32 -- Python 3.13.0, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
configfile: pytest.ini
collected 10 items

test\unit\calc_test.py .....                                             [ 80%]
test\unit\util_test.py ..                                                [100%]

- generated xml file:
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage\result-unit.xml -
============================== 10 passed in 0.17s ==============================
[Pipeline] sleep
Sleeping for 5 Seg
[Pipeline] junit
Grabando resultados de tests
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Static and Security Analysis)
[Pipeline] parallel
[Pipeline] { (Branch: Static)
[Pipeline] { (Branch: Security)
[Pipeline] stage
[Pipeline] { (Static)
```

```
[Pipeline] stage
[Pipeline] { (Security)
[Pipeline] node
Running on agent1
 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] node
Running on agent2
 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] {
[Pipeline] {
[Pipeline] bat
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>whoami
amaro\amaro

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>whoami
amaro\amaro
[Pipeline] bat
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>C:\Users\amaro\App
Data\Local\Programs\Python\Python313\Scripts\flake8.exe --exit-zero --format=pylint --exit-zero app  1>flake8.out

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>C:\Users\amaro\App
Data\Local\Programs\Python\Python313\Scripts\bandit.exe --exit-zero -r . -f custom -o bandit.out --msg-template "{abspath}:{line}: [{test_id}]
{msg}"
[main]      INFO        profile include tests: None
[main]      INFO        profile exclude tests: None
[main]      INFO        cli include tests: None
[main]      INFO        cli exclude tests: None
[main]      INFO        running on Python 3.13.0
[custom]    INFO        Result written to file: bandit.out
[Pipeline] catchError
[Pipeline] {
[Pipeline] recordIssues
WARNING: Unknown parameter(s) found for class type 'io.jenkins.plugins.analysis.core.util.WarningsQualityGate': healthy
[Bandit] Searching for all files in
'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage' that match the
pattern '**/bandit.out'
[Bandit] Traversing of symbolic links: enabled
```

[Bandit] -> found 1 file

[Bandit] Skipping file 'bandit.out' because it's empty

[Bandit] Skipping post processing

[Bandit] No filter has been set, publishing all 0 issues

[Bandit] Repository miner is not configured, skipping repository mining

[Bandit] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage' that match the pattern '**/bandit.out'

[Bandit] Traversing of symbolic links: enabled

[Bandit] -> found 1 file

[Bandit] Skipping file 'bandit.out' because it's empty

[Bandit] Skipping post processing

[Bandit] No filter has been set, publishing all 0 issues

[Bandit] Repository miner is not configured, skipping repository mining

[Bandit] Reference build recorder is not configured

[Bandit] No valid reference build found

[Bandit] All reported issues will be considered outstanding

[Bandit] Evaluating quality gates

[Bandit] -> All quality gates have been passed

[Bandit] -> Details for each quality gate:

[Bandit] - [Total (any severity)]: ≪Correcto≫ - (Actual value: 0, Quality gate: 2,00)

[Bandit] - [Total (any severity)]: ≪Correcto≫ - (Actual value: 0, Quality gate: 4,00)

[Bandit] Health report is disabled - skipping

[Bandit] Created analysis result for 0 issues (found 0 new issues, fixed 0 issues)

[Bandit] Attaching ResultAction with ID 'pylint' to build 'Unir/JENKINSFILE_agentes_feature_fix_coverage #5'.

[Checks API] No suitable checks publisher found.

[Pipeline] }

[Pipeline] // catchError

[Pipeline] }

[Pipeline] // node

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] recordIssues

WARNING: Unknown parameter(s) found for class type 'io.jenkins.plugins.analysis.core.util.WarningsQualityGate': healthy

[Flake8] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage' that match the pattern '**/flake8.out'

[Flake8] Traversing of symbolic links: enabled

[Flake8] -> found 1 file

[Flake8] Successfully parsed file C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage\flake8.out

[Flake8] -> found 9 issues (skipped 0 duplicates)

[Flake8] Successfully processed file 'flake8.out'

[Flake8] Post processing issues on 'agent1' with source code encoding 'windows-1252'

[Flake8] Creating SCM blamer to obtain author and commit information for affected files

[Flake8] -> No blamer installed yet. You need to install the 'git-forensics' plugin to enable blaming for Git.

[Flake8] Resolving file names for all issues in workspace 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage'

[Flake8] -> resolved paths in source directory (2 found, 0 not found)

[Flake8] Resolving module names from module definitions (build.xml, pom.xml, or Manifest.mf files)

[Flake8] -> resolved module names for 9 issues

[Flake8] Resolving package names (or namespaces) by parsing the affected files

[Flake8] -> resolved package names of 2 affected files

[Flake8] No filter has been set, publishing all 9 issues

[Flake8] Creating fingerprints for all affected code blocks to track issues over different builds

[Flake8] -> created fingerprints for 9 issues (skipped 0 issues)

[Flake8] Copying affected files to Jenkins' build folder 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes_feature_fix_coverage\builds\5\files-with-issues'

```
[Flake8] -> 2 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[Flake8] Skipping cleaning of source code files in old builds
[Flake8] Repository miner is not configured, skipping repository mining
[Flake8] Searching for all files in 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage' that match the pattern '**/flake8.out'
[Flake8] Traversing of symbolic links: enabled
[Flake8] -> found 1 file
[Flake8] Successfully parsed file C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage\flake8.out
[Flake8] -> found 9 issues (skipped 0 duplicates)
[Flake8] Successfully processed file 'flake8.out'
[Flake8] Post processing issues on 'agent1' with source code encoding 'windows-1252'
[Flake8] Creating SCM blamer to obtain author and commit information for affected files
[Flake8] -> No blamer installed yet. You need to install the 'git-forensics' plugin to enable blaming for Git.
[Flake8] Resolving file names for all issues in workspace 'C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage'
[Flake8] -> resolved paths in source directory (2 found, 0 not found)
[Flake8] Resolving module names from module definitions (build.xml, pom.xml, or Manifest.mf files)
[Flake8] -> resolved module names for 9 issues
[Flake8] Resolving package names (or namespaces) by parsing the affected files
[Flake8] -> resolved package names of 2 affected files
[Flake8] No filter has been set, publishing all 9 issues
[Flake8] Creating fingerprints for all affected code blocks to track issues over different builds
[Flake8] -> created fingerprints for 9 issues (skipped 0 issues)
[Flake8] Copying affected files to Jenkins' build folder 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes_feature_fix_coverage\builds\5\files-with-issues'
[Flake8] -> 2 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[Flake8] Skipping cleaning of source code files in old builds
[Flake8] Repository miner is not configured, skipping repository mining
[Flake8] Reference build recorder is not configured
[Flake8] No valid reference build found
[Flake8] All reported issues will be considered outstanding
[Flake8] Evaluating quality gates
[Flake8] -> Some quality gates have been missed: overall result is UNSTABLE
[Flake8] -> Details for each quality gate:
[Flake8]    - [Total (any severity)]: ≪Inestable≫ - (Actual value: 9, Quality gate: 8,00)
[Flake8]    - [Total (any severity)]: ≪Correcto≫ - (Actual value: 9, Quality gate: 10,00)
[Flake8] Health report is disabled - skipping
[Flake8] Created analysis result for 9 issues (found 0 new issues, fixed 0 issues)
[Flake8] Attaching ResultAction with ID 'flake8' to build 'Unir/JENKINSFILE_agentes_feature_fix_coverage #5'.
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // parallel
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Coverage)
[Pipeline] node
Running on agent1
 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] {
[Pipeline] bat
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>C:\Users\amaro\App
Data\Local\Programs\Python\Python313\Scripts\coverage.exe run --branch --source=app --omit=app\__init__.py,app\api.py -m pytest test\unit
============================= test session starts =============================
platform win32 -- Python 3.13.0, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
configfile: pytest.ini
collected 10 items

test\unit\calc_test.py .....                                             [ 80%]
test\unit\util_test.py ..                                                [100%]

============================== 10 passed in 0.06s ==============================

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>C:\Users\amaro\App
Data\Local\Programs\Python\Python313\Scripts\coverage.exe xml
Wrote XML report to coverage.xml
[Pipeline] catchError
[Pipeline] {
[Pipeline] cobertura
[Cobertura] Skipping Cobertura coverage report as build was not SUCCESS or better ...

[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Performance)
[Pipeline] node
Running on agent3
 in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>whoami
amaro\amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>hostname
Amaro
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>echo
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>SET
FLASK_APP=app\api.py

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>start                /B
C:\Users\amaro\AppData\Local\Programs\Python\Python313\python.exe -m flask run --host=0.0.0.0 --port=5000

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>timeout     /t     20
/nobreak  // Espera 20 segundos para asegurarse de que Flask estÃ© listo antes de correr las pruebas
ERROR: Sintaxis no v lida. La opci¢n predeterminada no est  permitida m s
de "1" veces.
Escriba "TIMEOUT /?" para su uso.

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage>C:\Users\amaro\Do
wnloads\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\jmeter -n -t test\jmeter\flask.jmx -f -l flask.jtl
Usage: python -m flask run [OPTIONS]
Try 'python -m flask run --help' for help.

Error: Could not import 'api'.
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
Creating summariser <summary>
Created the tree successfully using test\jmeter\flask.jmx
Starting standalone test @ 2025 Jan 23 17:19:11 CET (1737649151600)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary =    200 in 00:00:01 =  189.8/s Avg:    6 Min:    3 Max:    39 Err:    0 (0.00%)
Tidying up ...    @ 2025 Jan 23 17:19:12 CET (1737649152752)
... end of run
[Pipeline] script
[Pipeline] {
[Pipeline] perfReport
Creating parser with percentiles:'0,50,90,95,100,' filterRegex:null
Performance: Recording JMeterCsv reports '**/flask.jtl'
Performance:                JMeterCsv                copying                reports                to                master,                files
'[C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent3\workspace\Unir\JENKINSFILE_agentes_feature_fix_coverage\flask.jtl]'
Performance:                JMeterCsv                parsing                local                reports
'[C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes_feature_fix_coverage\builds\5\performance-
reports\JMeterCSV\flask.jtl]'
Performance:                Parsing                report                file
'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE_agentes_feature_fix_coverage\builds\5\performance-
reports\JMeterCSV\flask.jtl' with filterRegex 'null'.
Performance: No threshold configured for making the test unstable
Performance: No threshold configured for making the test failure
Performance: File flask.jtl reported 0.0% of errors [SUCCESS]. Build status is: UNSTABLE
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] End of Pipeline
Finished: UNSTABLE
```

- Explicación de cómo se ha hecho la mejora y por qué antes no se alcanzaba el 100% de cobertura.

1. Pruebas de la clase TestCalculate (Cálculos):

En los métodos de esta clase se realizan operaciones como la suma, la resta, la multiplicación, la división y la potenciación. Necesitamos asegurarnos de que todos los casos de excepciones y resultados posibles sean cubiertos.

- **Añadir más pruebas** para cubrir los casos en que el divisor es 0 en la división.
- **Agregar excepciones y otros valores no numéricos** en cada operación.

2. Pruebas de la clase TestUtil (Utilidades):

Para la función convert_to_number, además de los casos de valores válidos y errores con cadenas, también debemos cubrir valores adicionales que puedan ser convertidos a números, como los booleanos y los números en notación científica.