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:

**https://github.com/GiovannaLeon/helloworld/blob/master/CP2.1.1/cp2.1.1.txt**

pipeline {

agent any

environment {

PYTHON\_PATH = "C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\python.exe"

PYTHON\_PATH\_SCRIPTS = "C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\Scripts"

}

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%

%PYTHON\_PATH\_SCRIPTS%\\coverage.exe run --branch --source=app --omit=app\\\_\_init\_\_.py,app\\api.py -m pytest --junitxml=result-unit.xml test\\unit

'''

sleep(5) // Asegúrate de que los resultados estén listos

junit 'result-unit.xml' // Reporte de las pruebas unitarias

}

}

}

// Etapa de Cobertura de Pruebas (Reutilizando los resultados de Unit)

stage('Coverage') {

steps {

bat '''

%PYTHON\_PATH\_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 '''

%PYTHON\_PATH\_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 '''

%PYTHON\_PATH\_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 %PYTHON\_PATH% -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 {

// Leemos el archivo coverage.xml generado

// def coverageFile = readFile('../coverage.xml')

try {

def coverageFile = readFile('coverage.xml')

echo "Archivo coverage.xml leído correctamente."

// Expresión regular para extraer la cobertura de líneas

def lineCoverageMatch = (coverageFile =~ /<counter type="line" covered="(\d+)"/)

def lineCoverage = lineCoverageMatch ? lineCoverageMatch[0][1].toInteger() : 0

echo "Expresión 1."

// Expresión regular para extraer la cobertura de ramas

def branchCoverageMatch = (coverageFile =~ /<counter type="branch" covered="(\d+)" /)

def branchCoverage = branchCoverageMatch ? branchCoverageMatch[0][1].toInteger() : 0

echo "Expresión 2."

echo "Valor de lineCoverageMatch: ${lineCoverageMatch}"

echo "Valor de lineCoverage: ${lineCoverage}"

echo "Valor de branchCoverageMatch: ${branchCoverageMatch}"

echo "Valor de branchCoverage: ${branchCoverage}"

// Verificar cobertura por líneas

//if (lineCoverage < 85) {

// currentBuild.result = 'FAILURE'

//} else if (lineCoverage < 95) {

// currentBuild.result = 'UNSTABLE'

//} else {

// currentBuild.result = 'SUCCESS'

//}

//echo "Expresión 3."

// Verificar cobertura por ramas/condiciones

//if (branchCoverage < 80) {

// currentBuild.result = 'FAILURE'

//} else if (branchCoverage < 90) {

// currentBuild.result = 'UNSTABLE'

//} else {

// currentBuild.result = 'SUCCESS'

//}

//echo "Expresión 4."

} catch (Exception e) {

echo "Error al leer coverage.xml: ${e.message}"

}

// 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 agent2

in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\CP2.1.1

[Pipeline] {

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Get Code)

[Pipeline] git

The recommended git tool is: NONE

No credentials specified

Fetching changes from the remote Git repository

Checking out Revision a6361a5a1b8b84333ac998b3d5e0995a8e0dbd70 (refs/remotes/origin/master)

Commit message: "Add files via upload"

> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\CP2.1.1\.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 a6361a5a1b8b84333ac998b3d5e0995a8e0dbd70 # timeout=10

> git.exe branch -a -v --no-abbrev # timeout=10

> git.exe branch -D master # timeout=10

> git.exe checkout -b master a6361a5a1b8b84333ac998b3d5e0995a8e0dbd70 # timeout=10

> git.exe rev-list --no-walk a6361a5a1b8b84333ac998b3d5e0995a8e0dbd70 # timeout=10

[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\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\agents\agent2\workspace\Unir\CP2.1.1

25/01/2025 11:54 <DIR> .

24/01/2025 01:01 <DIR> ..

25/01/2025 11:54 53.248 .coverage

24/01/2025 00:40 40 .gitignore

24/01/2025 00:40 <DIR> .pytest\_cache

24/01/2025 00:40 <DIR> app

25/01/2025 11:54 426 bandit.out

25/01/2025 11:54 2.261 coverage.xml

25/01/2025 11:42 <DIR> CP2.1.1

25/01/2025 11:54 500 flake8.out

25/01/2025 11:54 19.938 flask.jtl

24/01/2025 00:40 <DIR> jenkinsFile\_1

24/01/2025 00:40 <DIR> jenkinsFile\_2

24/01/2025 00:40 <DIR> JenkinsFile\_3

24/01/2025 00:40 <DIR> jenkinsfile\_4

25/01/2025 11:54 10.493 jmeter.log

24/01/2025 00:40 175 pytest.ini

24/01/2025 00:40 418 README.md

25/01/2025 11:54 1.401 result-unit.xml

24/01/2025 00:40 <DIR> test

10 archivos 88.900 bytes

10 dirs 560.666.284.032 bytes libres

[Pipeline] echo

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\CP2.1.1

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Unit)

[Pipeline] catchError

[Pipeline] {

[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\CP2.1.1>SET PYTHONPATH=C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\CP2.1.1

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\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 --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\agent2\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\agents\agent2\workspace\Unir\CP2.1.1\result-unit.xml -

============================= 10 passed in 0.14s ==============================

[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\agents\agent2\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\agents\agent2\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\agents\agent2\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\agents\agent2\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 'agent2' 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\agent2\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\67\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\agent2\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\agents\agent2\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 'agent2' 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\agent2\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\67\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 #67'.

[Checks API] No suitable checks publisher found.

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Security)

[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\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\agents\agent2\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\agents\agent2\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 'agent2' 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\agents\agent2\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\67\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\agents\agent2\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\agents\agent2\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 'agent2' 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\agents\agent2\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\67\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 #67'.

[Checks API] No suitable checks publisher found.

[Pipeline] }

[Pipeline] // catchError

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Performance)

[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\CP2.1.1>SET FLASK\_APP=app\api.py

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\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\agents\agent2\workspace\Unir\CP2.1.1>timeout /t 10 /nobreak

ERROR: No es compatible la redirecci¢n de entradas, saliendo inmediatamente

del proceso.

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\workspace\Unir\CP2.1.1>rem Espera 10 segundos para asegurarse de que Flask estÃ© listo antes de correr las pruebas

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\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://192.168.1.57:5000

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 25 11:56:35 CET (1737802595710)

Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445

summary + 1 in 00:00:00 = 7.2/s Avg: 25 Min: 25 Max: 25 Err: 0 (0.00%) Active: 3 Started: 3 Finished: 0

summary + 159 in 00:00:01 = 187.5/s Avg: 2 Min: 1 Max: 25 Err: 0 (0.00%) Active: 0 Started: 20 Finished: 20

summary = 160 in 00:00:01 = 162.3/s Avg: 2 Min: 1 Max: 25 Err: 0 (0.00%)

Tidying up ... @ 2025 Jan 25 11:56:36 CET (1737802596771)

... end of run

[Pipeline] script

[Pipeline] {

[Pipeline] readFile

[Pipeline] echo

Archivo coverage.xml leído correctamente.

[Pipeline] echo

Expresión 1.

[Pipeline] echo

Expresión 2.

[Pipeline] echo

Valor de lineCoverageMatch: java.util.regex.Matcher[pattern=<counter type="line" covered="(\d+)" region=0,2261 lastmatch=]

[Pipeline] echo

Valor de lineCoverage: 0

[Pipeline] echo

Valor de branchCoverageMatch: java.util.regex.Matcher[pattern=<counter type="branch" covered="(\d+)" region=0,2261 lastmatch=]

[Pipeline] echo

Valor de branchCoverage: 0

[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\agents\agent2\workspace\Unir\CP2.1.1\flask.jtl]'

Performance: JMeterCsv parsing local reports '[C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\CP2.1.1\builds\67\performance-reports\JMeterCSV\flask.jtl]'

Performance: Parsing report file 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\CP2.1.1\builds\67\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] // withEnv

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: UNSTABLE

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

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Resultados

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Captura de pantalla de Jenkins donde se vea el resultado de los plugins:
  + Plugin Junit: evolución de los resultados de las pruebas unitarias.
  + Interfaz de usuario gráfica

    Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

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

Interfaz de usuario gráfica

Descripción generada automáticamente

* + Plugin Warnings-NG (Flake8): evolución de los hallazgos encontrados.
  + Gráfico, Gráfico de líneas

    Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

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

    Descripción generada automáticamente

Gráfico

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* + 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.

Aplicación

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

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

* Explicación del funcionamiento del pipeline.

**Estructura general del Pipeline:**

1. **Agente**:

agent any

Esto significa que el pipeline puede ejecutarse en cualquier agente disponible. Jenkins utilizará el agente disponible para ejecutar el pipeline.

1. **Entorno**:

environment {

PYTHON\_PATH = "C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\python.exe"

PYTHON\_PATH\_SCRIPTS = "C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\Scripts"

}

En esta sección se definen variables de entorno para la ruta de Python y su directorio de scripts, de forma que puedas ejecutar los comandos de Python desde cualquier lugar del pipeline.

**Etapas del Pipeline:**

**1. Get Code:**

stage('Get Code') {

steps {

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

bat "dir"

echo WORKSPACE

}

}

* **git**: Clona el repositorio desde GitHub.
* **bat "dir"**: Muestra el listado de archivos en el directorio.
* **echo WORKSPACE**: Muestra el espacio de trabajo actual de Jenkins.

**2. Unit:**

Esta etapa ejecuta las pruebas unitarias utilizando pytest y genera un reporte XML de las pruebas.

stage('Unit') {

steps {

catchError(buildResult: 'UNSTABLE', stageResult: 'FAILURE') {

bat '''

SET PYTHONPATH=%WORKSPACE%

%PYTHON\_PATH\_SCRIPTS%\\coverage.exe run --branch --source=app --omit=app\\\_\_init\_\_.py,app\\api.py -m pytest --junitxml=result-unit.xml test\\unit

'''

sleep(5)

junit 'result-unit.xml' // Reporte de las pruebas unitarias

}

}

}

* Ejecuta pytest con cobertura de código utilizando coverage.exe y genera un reporte en XML (result-unit.xml).
* Usa junit para publicar los resultados de las pruebas unitarias a Jenkins.

**3. Coverage:**

Esta etapa utiliza los resultados de las pruebas unitarias para generar el reporte de cobertura y publica este reporte en Jenkins.

stage('Coverage') {

steps {

bat '''

%PYTHON\_PATH\_SCRIPTS%\\coverage.exe xml

'''

catchError(buildResult: 'UNSTABLE', stageResult: 'FAILURE') {

cobertura coberturaReportFile: '\*\*/coverage.xml', conditionalCoverageTargets: '100,0,80', lineCoverageTargets: '100,0,90'

}

}

}

* Usa coverage.exe para generar el archivo coverage.xml, que luego es utilizado por el plugin de cobertura de Jenkins (cobertura) para mostrar un reporte.
* Los umbrales para la cobertura de código están definidos con conditionalCoverageTargets y lineCoverageTargets.

**4. Static:**

En esta etapa se realiza el análisis estático de código utilizando flake8 para detectar problemas de estilo de código.

stage('Static') {

steps {

bat '''

%PYTHON\_PATH\_SCRIPTS%\\flake8.exe --exit-zero --format=pylint --exit-zero app >flake8.out

'''

recordIssues tools: [flake8(name: 'Flake8', pattern: '\*\*/flake8.out')],

qualityGates: [

[threshold: 8, type: 'TOTAL', unstable: true],

[threshold: 10, type: 'TOTAL', unstable: false, healthy: false]

]

}

}

* Ejecuta flake8 para verificar los problemas de estilo de código.
* Publica los resultados con recordIssues y aplica umbrales de calidad.

**5. Security:**

Aquí se realiza un análisis de seguridad utilizando bandit, una herramienta para detectar vulnerabilidades en el código.

stage('Security') {

steps {

bat '''

%PYTHON\_PATH\_SCRIPTS%\\bandit.exe --exit-zero -r . -f custom -o bandit.out --msg-template "{abspath}:{line}: [{test\_id}] {msg}"

'''

catchError(buildResult: 'SUCCESS', stageResult: 'UNSTABLE') {

recordIssues tools: [pyLint(name: 'Bandit', pattern: '\*\*/bandit.out')],

qualityGates: [

[threshold: 2, type: 'TOTAL', unstable: true],

[threshold: 4, type: 'TOTAL', unstable: false, healthy: false]

]

}

}

}

* Utiliza bandit para analizar el código en busca de vulnerabilidades de seguridad.
* Publica los resultados con el plugin de Jenkins para mostrar los problemas de seguridad encontrados.

**6. Performance:**

Esta etapa realiza pruebas de rendimiento utilizando JMeter y ejecuta una aplicación Flask para simular el comportamiento de la aplicación bajo prueba.

stage('Performance') {

steps {

bat '''

SET FLASK\_APP=app\\api.py

start /B %PYTHON\_PATH% -m flask run --host=0.0.0.0 --port=5000

timeout /t 10 /nobreak

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 {

try {

def coverageFile = readFile('coverage.xml')

echo "Archivo coverage.xml leído correctamente."

// Procesamiento de cobertura

} catch (Exception e) {

echo "Error al leer coverage.xml: ${e.message}"

}

}

catchError(buildResult: 'SUCCESS', stageResult: 'UNSTABLE') {

perfReport sourceDataFiles: '\*\*/flask.jtl'

}

}

}

* Ejecuta Flask en un servidor local y luego ejecuta las pruebas de rendimiento con JMeter.
* Utiliza el reporte generado por JMeter (flask.jtl) para mostrar el rendimiento de la aplicación.

**Resumen:**

Este pipeline de Jenkins realiza un ciclo completo de integración continua (CI) para una aplicación Python. Abarca:

1. **Obtener el código** desde un repositorio Git.
2. **Ejecutar pruebas unitarias** y generar reportes de resultados.
3. **Generar reportes de cobertura de código** y mostrar el resultado en Jenkins.
4. **Realizar análisis estático** de código con flake8.
5. **Realizar un análisis de seguridad** con bandit.
6. **Realizar pruebas de rendimiento** con JMeter.

Este flujo asegura que el código sea revisado desde varios puntos de vista (funcionalidad, estilo, seguridad, y rendimiento) antes de ser desplegado, lo que facilita la integración continua y mejora la calidad del software.

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.

**https://github.com/GiovannaLeon/helloworld/blob/master/CP2.1.1/JENKINSFILE\_agentes.txt**

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

pipeline {

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

pipeline {

agent none

environment {

PYTHON\_PATH = "C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\python.exe"

PYTHON\_PATH\_SCRIPTS = "C:\\Users\\amaro\\AppData\\Local\\Programs\\Python\\Python313\\Scripts"

}

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%

%PYTHON\_PATH\_SCRIPTS%\\coverage.exe run --branch --source=app --omit=app\\\_\_init\_\_.py,app\\api.py -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 '''

%PYTHON\_PATH\_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 '''

%PYTHON\_PATH\_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 '''

%PYTHON\_PATH\_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 %PYTHON\_PATH% -m flask run --host=0.0.0.0 --port=5000

timeout /t 20 /nobreak // 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 {

// 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).

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico, Teams

Descripción generada automáticamente

Lanzada por el usuario Giovanna leon

[Pipeline] Start of Pipeline

[Pipeline] withEnv

[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 a6361a5a1b8b84333ac998b3d5e0995a8e0dbd70 (refs/remotes/origin/master)

> 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 a6361a5a1b8b84333ac998b3d5e0995a8e0dbd70 # timeout=10

> git.exe branch -a -v --no-abbrev # timeout=10

> git.exe branch -D master # timeout=10

> git.exe checkout -b master a6361a5a1b8b84333ac998b3d5e0995a8e0dbd70 # timeout=10

Commit message: "Add files via upload"

> git.exe rev-list --no-walk a6361a5a1b8b84333ac998b3d5e0995a8e0dbd70 # 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

25/01/2025 12:00 <DIR> .

23/01/2025 16:51 <DIR> ..

25/01/2025 12:00 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

25/01/2025 12:00 2.273 coverage.xml

25/01/2025 12:00 <DIR> CP2.1.1

25/01/2025 12:00 500 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

25/01/2025 12:00 1.401 result-unit.xml

20/01/2025 01:38 <DIR> test

7 archivos 58.055 bytes

10 dirs 560.652.320.768 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\Scripts\coverage.exe run --branch --source=app --omit=app\\_\_init\_\_.py,app\api.py -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 0.14s ==============================

[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 #21'.

[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\21\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\21\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 #21'.

[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 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 25 12:01:56 CET (1737802916113)

Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445

summary = 200 in 00:00:01 = 223.5/s Avg: 1 Min: 1 Max: 19 Err: 0 (0.00%)

Tidying up ... @ 2025 Jan 25 12:01:57 CET (1737802917089)

... 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\21\performance-reports\JMeterCSV\flask.jtl]'

Performance: Parsing report file 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE\_agentes\builds\21\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] }

[Pipeline] // withEnv

[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](http://localhost:8080/user/giovanna)

[Pipeline] Start of Pipeline

[Pipeline] stage

[Pipeline] { (Get Code)

[Pipeline] node

Running on [agent1](http://localhost:8080/computer/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](http://localhost:8080/computer/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](http://localhost:8080/computer/agent1/) in C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent1\workspace\Unir\JENKINSFILE\_agentes

[Pipeline] node

Running on [agent2](http://localhost:8080/computer/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](http://localhost:8080/computer/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](http://localhost:8080/computer/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”.

**https://github.com/GiovannaLeon/helloworld/blob/master/CP2.1.1/JENKINSFILE\_agentes\_master.txt**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* 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] withEnv

[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

> 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

Checking out Revision a6361a5a1b8b84333ac998b3d5e0995a8e0dbd70 (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 a6361a5a1b8b84333ac998b3d5e0995a8e0dbd70 # timeout=10

> git.exe branch -a -v --no-abbrev # timeout=10

> git.exe branch -D master # timeout=10

> git.exe checkout -b master a6361a5a1b8b84333ac998b3d5e0995a8e0dbd70 # 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

25/01/2025 12:06 <DIR> .

23/01/2025 16:51 <DIR> ..

23/01/2025 17:04 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 17:04 2.280 coverage.xml

25/01/2025 12:06 <DIR> CP2.1.1

23/01/2025 17:04 500 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 17:04 1.401 result-unit.xml

23/01/2025 16:51 <DIR> test

7 archivos 58.062 bytes

10 dirs 560.655.937.536 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\Programs\Python\Python313\Scripts\coverage.exe run --branch --source=app --omit=app\\_\_init\_\_.py,app\api.py -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 0.10s ==============================

[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\_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\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\_master>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\_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 #9'.

[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\_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\9\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\9\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 #9'.

[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\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\_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\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 25 12:06:39 CET (1737803199381)

Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445

summary = 200 in 00:00:01 = 225.0/s Avg: 1 Min: 1 Max: 19 Err: 0 (0.00%)

Tidying up ... @ 2025 Jan 25 12:06:40 CET (1737803200343)

... 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\9\performance-reports\JMeterCSV\flask.jtl]'

Performance: Parsing report file 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE\_agentes\_master\builds\9\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] }

[Pipeline] // withEnv

[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.

**https://github.com/GiovannaLeon/helloworld/blob/master/CP2.1.1/JENKINSFILE\_agentes\_feature\_fix\_coverage.txt**

Lanzada por el usuario Giovanna leon

[Pipeline] Start of Pipeline

[Pipeline] withEnv

[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 5408c6c0d7ec57cd1c5391bf5f2e6b9c1d5f43ab (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

Commit message: "Update calc\_test.py"

> git.exe checkout -f 5408c6c0d7ec57cd1c5391bf5f2e6b9c1d5f43ab # timeout=10

> 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 5408c6c0d7ec57cd1c5391bf5f2e6b9c1d5f43ab # timeout=10

> git.exe rev-list --no-walk 2175ef423cb0425c8e7cb63fe4e6f30cbc6a4e44 # 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

25/01/2025 12:29 <DIR> .

23/01/2025 16:51 <DIR> ..

25/01/2025 12:29 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

25/01/2025 12:29 2.245 coverage.xml

25/01/2025 12:29 500 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

25/01/2025 12:29 2.818 result-unit.xml

20/01/2025 23:58 <DIR> test

7 archivos 59.444 bytes

9 dirs 560.655.790.080 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\AppData\Local\Programs\Python\Python313\Scripts\coverage.exe run --branch --source=app --omit=app\\_\_init\_\_.py,app\api.py -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 9 items

test\unit\calc\_test.py ....... [ 77%]

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 -

============================== 9 passed in 0.13s ==============================

[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

[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\agents\agent2\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\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\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\_feature\_fix\_coverage>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\_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 #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\_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\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\_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\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\_feature\_fix\_coverage #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\_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\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\_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\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 25 12:33:13 CET (1737804793513)

Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445

summary = 200 in 00:00:01 = 225.2/s Avg: 1 Min: 1 Max: 18 Err: 0 (0.00%)

Tidying up ... @ 2025 Jan 25 12:33:14 CET (1737804794471)

... 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\13\performance-reports\JMeterCSV\flask.jtl]'

Performance: Parsing report file 'C:\ProgramData\Jenkins\.jenkins\jobs\Unir\jobs\JENKINSFILE\_agentes\_feature\_fix\_coverage\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] }

[Pipeline] // withEnv

[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.

***He tenido problemas a la vez que he agregado para la rama*** [***feature\_fix\_coverage***](https://github.com/GiovannaLeon/helloworld/tree/feature_fix_coverage)***, calc\_ y util\_, no he alcanzado el 100% en calc, pero si en útil\_rest.***