

DevOps e Integración Continua

PRÁCTICA 2 - Creación de una librería compartida de Jenkins

Presentado a: **Javier Gutiérrez Navío**
Presentado Por: **Diana Cristy Florez Bravo**

Objetivo

El objetivo de esta práctica es crear una “Shared Lib” de Jenkins propia, y usarla en un pipeline. Como conocimientos a demostrar será:

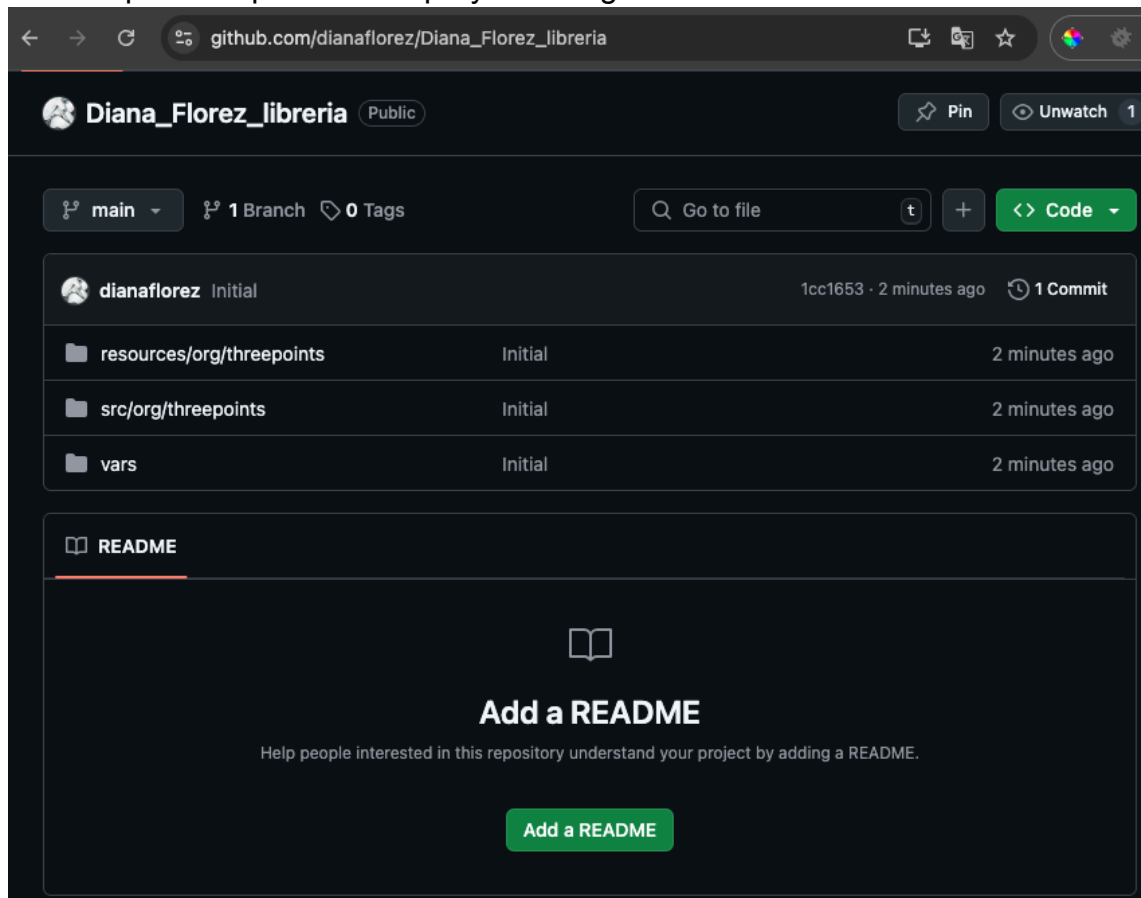
- Creación de la estructura de una librería básica de Jenkins.
- Uso de una librería en un pipeline declarativo.

1 – Creación de una librería de Jenkins en Github

Se debe de crear un proyecto en Github, llamado, “_libreria”, que contenga una estructura vacía con las 3 carpetas indicadas en el curso (/vars, /src/org/threepoints y /resources/org/threepoints), con un archivo vacío llamado “.placeholder” en cada una.

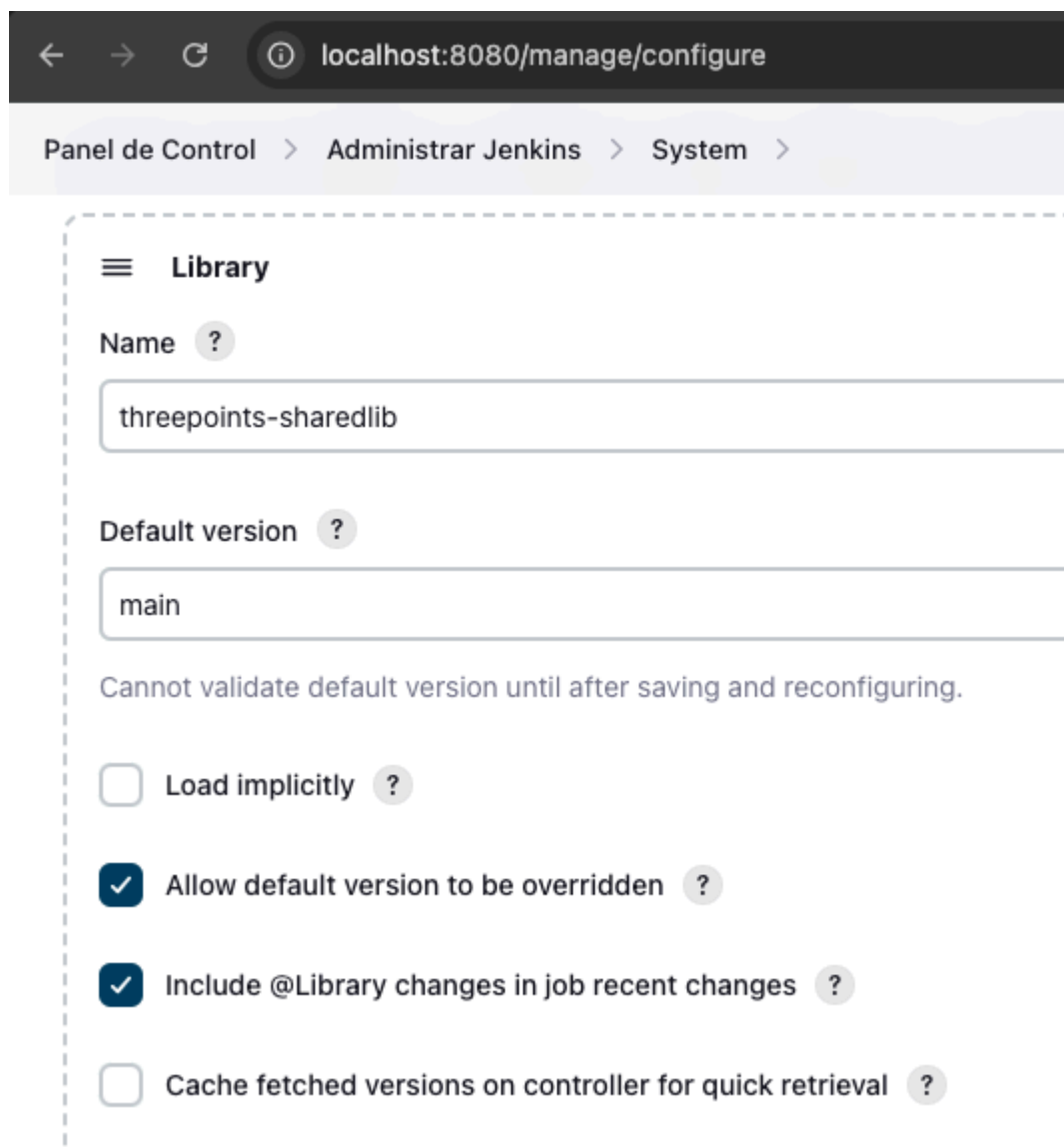
Los entregables de este ejercicio serán:

- Un zip con el proyecto.
- Una captura de pantalla del proyecto de git.



2 – Inclusión de la librería en Jenkins de forma explícita (1 puntos)

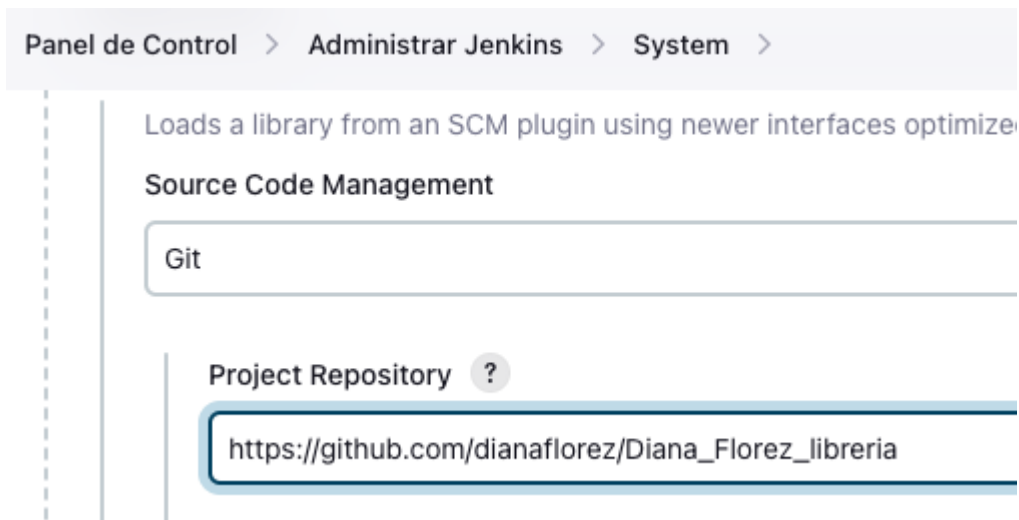
En este ejercicio, vamos a configurar Jenkins para poder leer la librería compartida que hemos creado en el ejercicio anterior. Para ello, debemos de seguir las instrucciones dadas en clase para poder asociar en el Jenkins de la máquina virtual, nuestra librería, bajo el nombre `threepointssharedlib`. Al asociar la librería, se debe de desactivar la opción de “Load implicitly”, para permitir cargar la librería posteriormente de la forma que se indican en las diapositivas, implícitamente. Durante este ejercicio, se debe de entregar una captura de pantalla con la configuración de Jenkins.



The screenshot shows the Jenkins web interface at `localhost:8080/manage/configure`. The breadcrumb navigation is `Panel de Control > Administrar Jenkins > System >`. The main section is titled **Library**. It contains the following fields and options:

- Name**: A text input field containing `threepoints-sharedlib`.
- Default version**: A text input field containing `main`.
- A message below the version field: `Cannot validate default version until after saving and reconfiguring.`
- Load implicitly**: An unchecked checkbox.
- Allow default version to be overridden**: A checked checkbox.
- Include @Library changes in job recent changes**: A checked checkbox.
- Cache fetched versions on controller for quick retrieval**: An unchecked checkbox.

- Agregamos el nombre del repositorio de git



3 – Programación de la librería para que englobe el análisis de código e inclusión en el pipeline

Durante esta práctica, para normalizar el cómo se realizan los análisis de código, e incluirlo siempre de forma sencilla, vamos a crear un script dentro de nuestra JSL que nos permita usarlo siempre de la misma forma. El script se debe de ubicar en “vars” en nuestra librería, y llamarse “staticAnalysis.groovy”, el cual debe de implementar una función “call” que permita ejecutar un análisis estático de código, y esperar durante 5m con un timeout al resultado. Debe de poder recibir un parámetro booleano el cual determine tras evaluar el QualityGate si debe de abortar el pipeline o no (por defecto valdrá “false”, lo cual significa que no debe de abortar el pipeline). Debido a que no usamos en la parte obligatoria SonarQube, la ejecución del escaneo de SonarQube debe de ser sustituida por un sh ‘echo “Ejecución de las pruebas de calidad de código”’, pero manteniendo el sonarenv y el timeout. En este ejercicio, no hay que usar SonarQube.

Así mismo, deberá recibir otro parámetro adicional, con el cual deberá abortar el pipeline según el parámetro que reciba, siendo este un booleano llamado “abortPipeline”. Si es True, abortará, si es False, continuará el pipeline. Para tener la puntuación perfecta de este ejercicio, se debe de mostrar además su utilización en un pipeline declarativo, aconsejando hacerlo con el de la práctica 2. Como entregable de este ejercicio se debe de dar:

- La librería comprimida en un zip, bajo el nombre devops_practica2_ejercicio3.zip.
- Una captura de pantalla de un pipeline en ejecución.
- El código del pipeline que use esta función.
- Capturas de pantalla de la configuración del Job de Jenkins para poder reproducirlo.

```
ceholder resources/...  .placeholder src/...  .placeholder vars  staticAnalysis.groovy U X  EXPLORER  ...  
vars > staticAnalysis.groovy  
1  def call(boolean abortOnQualityGate = false, boolean abortPipeline = false) {  
2      echo "Ejecución de las pruebas de calidad de código"  
3  
4      withSonarQubeEnv('SonarQube') {  
5          sh 'echo "Simulando ejecución de análisis estático..."'  
6      }  
7  
8      timeout(time: 5, unit: 'MINUTES') {  
9          echo "Simulación de evaluación del Quality Gate"  
10  
11         if (abortOnQualityGate) {  
12             echo "Quality Gate ha fallado. Abortando pipeline..."  
13             error("Pipeline abortado debido a fallo en Quality Gate.")  
14         }  
15     }  
16  
17     // Si el parámetro 'abortPipeline' es true, abortar el pipeline manualmente  
18     if (abortPipeline) {  
19         echo "Abortando pipeline según la configuración..."  
20         error("Pipeline abortado manualmente según el parámetro.")  
21     }  
22  
23     echo "Análisis de código completado correctamente."  
24 }  
25
```

- Realice varias pruebas con el pipeline

1. Prueba staticAnalysis(false, false). Me di cuenta que en esta opción sale en verde el resultado.

Definition

Pipeline script

Script ?

```
1  
2  library 'threepoints-sharedlib'  
3  
4  pipeline {  
5      agent any  
6      stages {  
7          stage('Static Analysis') {  
8              steps {  
9                  staticAnalysis(false, false)  
10             }  
11         }  
12     }  
13 }  
14  
15
```

Status

Pipeline Pipeline_Static_Analysis

</> Changes

▶ Construir ahora

⚙ Configurar

🗑 Borrar Pipeline

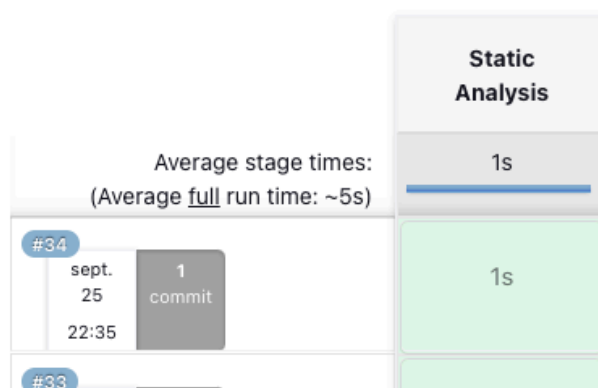
🔍 Full Stage View

🌊 Open Blue Ocean

✎ Rename

❓ Pipeline Syntax

Stage View



☁ Historia de tareas **Tendencia** ▼

Panel de Control > Pipeline_Static_Analysis > #34

Status

✓ Salida de consola

</> Changes

📄 Console Output

📄 View as plain text

✎ Edit Build Information

🗑 Delete build '#34'

🔍 Git Build Data

🌊 Open Blue Ocean

🔄 Restart from Stage

🔄 Replay

📋 Pipeline Steps

📁 Workspaces

← Previous Build

```
Lanzada por el usuario threepoints
[Pipeline] Start of Pipeline
[Pipeline] library
Loading library threepoints-sharedlib@main
Attempting to resolve main from remote references...
> git --version # timeout=10
> git --version # 'git version 2.39.2'
> git ls-remote -h -- http://github.com/dianaflomez/Diana_Florez_libreria # timeout=10
Found match: refs/heads/main revision c303ffcd1e9041c5fcca2e3e8d1797631b0d1380
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir
/var/lib/jenkins/workspace/Pipeline_Static_Analysis@libs/95f38eb155b90289d8c37e7a16c44ba66c071dd
f9abe335e583ed26bded8c39b/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url http://github.com/dianaflomez/Diana_Florez_libreria # timeout=10
Fetching without tags
Fetching upstream changes from http://github.com/dianaflomez/Diana_Florez_libreria
> git --version # timeout=10
> git --version # 'git version 2.39.2'
> git fetch --no-tags --progress -- http://github.com/dianaflomez/Diana_Florez_libreria
+refs/heads/*:refs/remotes/origin/* # timeout=10
```

```

> git --version # 'git version 2.39.2'
> git fetch --no-tags --force --progress -- http://github.com/dianaflorez/Diana_Florez_libreria +refs/heads/*:refs/remotes/* # timeout=10
Checking out Revision c303ffcd1e9041c5fcca2e3e8d1797631b0d1380 (main)
> git config core.sparsecheckout # timeout=10
> git checkout -f c303ffcd1e9041c5fcca2e3e8d1797631b0d1380 # timeout=10
Commit message: "[test]"
> git rev-list --no-walk 2b173c8e4c323efab4f005e9dee5777abd50367d # timeout=10
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Pipeline_Static_Analysis
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Static Analysis)
[Pipeline] echo
Ejecución de las pruebas de calidad de código
[Pipeline] sh
+ echo Simulando ejecución de análisis estático...
Simulando ejecución de análisis estático...
[Pipeline] timeout
Timeout set to expire in 5 Min 0 Seg
[Pipeline] {
[Pipeline] echo
Simulación de evaluación del Quality Gate
[Pipeline] }
[Pipeline] // timeout
[Pipeline] echo
Análisis de código completado correctamente.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

2. Prueba staticAnalysisus(false, true)

Definition


Pipeline script

Script ?


```

1
2 library 'threepoints-sharedlib'
3
4 pipeline {
5     agent any
6     stages {
7         stage('Static Analysis') {
8             steps {
9                 staticAnalysis(false,true)
10            }
11        }
12    }
13 }
14
15


```

 Status


 Changes

 Construir ahora

 Configurar

 Borrar Pipeline

 Full Stage View

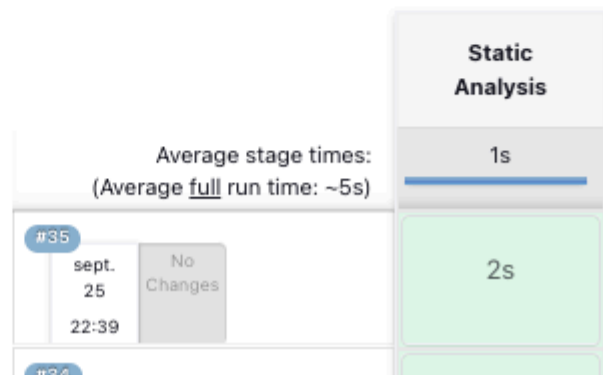
 Open Blue Ocean


 Rename

 Pipeline Syntax

Pipeline Pipeline_Static_Analysis

Stage View





 Historia de tareas **Tendencia** ▼

Panel de Control > Pipeline_Static_Analysis > #35


 Status

 Changes

 Console Output


 View as plain text

 Edit Build Information

 Delete build '#35'

 Git Build Data

 Open Blue Ocean

 Restart from Stage

 Replay

 Pipeline Steps

 Workspaces

 Previous Build

✓ Salida de consola

```
Lanzada por el usuario threepoints
[Pipeline] Start of Pipeline
[Pipeline] library
Loading library threepoints-sharedlib@main
Attempting to resolve main from remote references...
> git --version # timeout=10
> git --version # 'git version 2.39.2'
> git ls-remote -h -- http://github.com/dianaflomez/Diana_Flores_libreria # timeout=10
Found match: refs/heads/main revision c303ffcd1e9041c5fcca2e3e8d1797631b0d1380
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir
/var/lib/jenkins/workspace/Pipeline_Static_Analysis@libs/95f38eb155b90289d8c37e7a16c44ba66c071ddf9abe335e583ed26bde8c39b/.git #
timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url http://github.com/dianaflomez/Diana_Flores_libreria # timeout=10
Fetching without tags
Fetching upstream changes from http://github.com/dianaflomez/Diana_Flores_libreria
> git --version # timeout=10
> git --version # 'git version 2.39.2'
> git fetch --no-tags --force --progress -- http://github.com/dianaflomez/Diana_Flores_libreria +refs/heads/*:refs/remotes/origin/*
# timeout=10
Checking out Revision c303ffcd1e9041c5fcca2e3e8d1797631b0d1380 (main)
> git config core.sparsecheckout # timeout=10
> git checkout -f c303ffcd1e9041c5fcca2e3e8d1797631b0d1380 # timeout=10
Commit message: "[test]"
> git rev-list --no-walk c303ffcd1e9041c5fcca2e3e8d1797631b0d1380 # timeout=10
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Pipeline_Static_Analysis
[Pipeline]
```

4 – Configurar el pipeline para ser dependiente de las variables de entorno

En este último ejercicio, vamos a ampliar la librería del ejercicio 3 para ver cómo usar variables de entorno genéricas de Jenkins. La función de `staticAnalysis.groovy`, anteriormente definida, debe de leer de una variable de entorno (directamente o pasada como parámetro desde el pipeline) y determinar según el nombre de la rama de la que venga, si debe de cortar el pipeline si falla el qualityGate. Debe de implementar la siguiente heurística: - Si el argumento pasado en el ejercicio 3 está a True, lo debe de cortar siempre. - Si no, evaluará el nombre de la rama de git de la que proviene la ejecución y: o Si es igual a "master", debe de cortar el pipeline. o Si el nombre de la rama empieza por "hotfix", debe de cortar el pipeline. o Si el nombre vale cualquier otra cosa, no debería cortarlo. Se pueden encontrar ayuda sobre las variables globales en los apuntes de clase y en los siguientes enlaces:

- Guía oficial: <https://www.jenkins.io/doc/book/pipeline/jenkinsfile/#using-environmentvariables>
- Como entregable de este ejercicio se debe de dar:
 - La librería comprimida en un zip, bajo el nombre devops_practica2_ejercicio4.zip.
 - Una captura de pantalla de un pipeline en ejecución.
 - El código del pipeline que use esta función.
 - Capturas de pantalla de la configuración del Job de Jenkins para poder reproducirlo.

```
rs > staticAnalysis.groovy
1  def call(boolean abortOnQualityGate = false, boolean abortPipeline = false) {
2
3      def branchName = env.BRANCH_NAME ?: 'unknown'
4
5      if (abortPipeline || branchName == 'master' || branchName.startsWith('hotfix')) {
6          echo "Branch: ${branchName}. Aborting pipeline due to branch rules."
7          error("Pipeline aborted due to branch rules.")
8      }
9
10     echo "Ejecución de las pruebas de calidad de código"
11
12     sh 'echo "Ejecución de las pruebas de calidad de código"'
13
14     timeout(time: 5, unit: 'MINUTES') {
15         // Aquí iría la llamada a SonarQube, pero en este caso simulamos la ejecución
16         sh 'echo "Esperando el resultado del análisis estático"'
17     }
18
19     if (abortOnQualityGate) {
20         echo "Quality Gate failed. Aborting pipeline."
21         error("Quality Gate failed. Pipeline aborted.")
22     } else {
23         echo "Quality Gate passed. Continuing pipeline."
24     }
25
26 }
27
```


Pipeline

Definition

Pipeline script

Script ?

```
1
2 library 'threepoints-sharedlib'
3
4 pipeline {
5     agent any
6     environment {
7         BRANCH_NAME = "${env.BRANCH_NAME}"
8     }
9
10    stages {
11        stage('Static Analysis') {
12            steps {
13                staticAnalysis(false, true)
14            }
15        }
16    }
17
18 }
19
```



Status



Changes



Construir ahora



Configurar



Borrar Pipeline



Full Stage View



Open Blue Ocean



Rename



Pipeline Syntax



Historia de tareas

Tendencia ▾

Pipeline Pipeline_Static_Analysis

Stage View



✓ Salida de consola

```
Lanzada por el usuario threepoints
[Pipeline] Start of Pipeline
[Pipeline] library
Loading library threepoints-sharedlib@main
Attempting to resolve main from remote references...
> git --version # timeout=10
> git --version # 'git version 2.39.2'
> git ls-remote -h -- http://github.com/dianaflomez/Diana_Florez_libreria # timeout=10
Found match: refs/heads/main revision c303ffcd1e9041c5fcca2e3e8d1797631b0d1380
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Pipeline_Static_Analysis@libs/95f38eb155b90289d8c37e7a16c44ba66c071ddf9abe335e583ed26bded8c39b/.git #
timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url http://github.com/dianaflomez/Diana_Florez_libreria # timeout=10
Fetching without tags
Fetching upstream changes from http://github.com/dianaflomez/Diana_Florez_libreria
> git --version # timeout=10
> git --version # 'git version 2.39.2'
> git fetch --no-tags --force --progress -- http://github.com/dianaflomez/Diana_Florez_libreria +refs/heads/*:refs/remotes/origin/* # timeout=10
Checking out Revision c303ffcd1e9041c5fcca2e3e8d1797631b0d1380 (main)
> git config core.sparsecheckout # timeout=10
> git checkout -f c303ffcd1e9041c5fcca2e3e8d1797631b0d1380 # timeout=10
Commit message: "[test]"
> git rev-list --no-walk c303ffcd1e9041c5fcca2e3e8d1797631b0d1380 # timeout=10
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Pipeline_Static_Analysis
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Static Analysis)
[Pipeline] echo
Ejecución de las pruebas de calidad de código
```

```
[Pipeline] echo
Ejecución de las pruebas de calidad de código
[Pipeline] sh
+ echo Simulando ejecución de análisis estático...
Simulando ejecución de análisis estático...
[Pipeline] timeout
Timeout set to expire in 5 Min 0 Seg
[Pipeline] {
[Pipeline] echo
Simulación de evaluación del Quality Gate
[Pipeline] }
[Pipeline] // timeout
[Pipeline] echo
Abortando pipeline según la configuración...
[Pipeline] echo
Análisis de código completado correctamente.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

