

## GESTIONAR USUARIOS EN UN SISTEMA LINUX A TRAVÉS DE PIPELINE DECLARATIVO EN JENKINS

- ***Para poder crear y eliminar usuarios en un sistema Linux desde Jenkins con un pipeline declarativo, es necesario entender como integrar varias herramientas y tecnologías. A continuación se detallan los pasos a seguir***

### Integración de herramientas y tecnologías:

#### ➤ 1. Jenkins

- Jenkins es una plataforma ideal para automatizar pipeline declarativos debido a su arquitectura versátil y estandarizada.

#### Algunos de sus beneficios:

**A.** integración (CI)y entrega (CC)continua: desde aquí se controlaran la creación y eliminación de usuarios, conectándose a maquinas de Linux para ejecutar los scripts de gestión de usuarios.

**B.** Escalabilidad y rendimiento

#### ➤ Github

- Es una plataforma de desarrollo colaborativo para proyectos de software. Ofrece variedad de herramientas y características que facilitan la colaboración, la gestión de versiones y la entrega de proyectos.

### INTEGRACIÓN: JENKINS CON GITHUB:

- Permite automatizar la construcción , prueba y despliegue de proyectos de software alojados en Github.

#### REQUISITOS:

- a. cuenta de Github
  - b. instancia de Jenkins
  - c. instalar Plug-in de Github en Jenkins
- Esto permite que Jenkins recupere el código fuente automáticamente cuando haya cambios en el repositorio

### USO DE GITHUB

- Alojara el código fuente del pipeline y los script necesarios para la creación y eliminación de usuarios.

### 3. BASH SCRIPT:

- **Uso:** los scripts de bash se ejecutaran en las maquinas de Linux para gestionar la creación y eliminación de usuarios directamente en el sistema de Linux.
- **Integración:** Github alojara en sus repositorios los script, que se ejecutaran en las maquinas destino a través de comandos enviados desde Jenkins.

#### 4. Email para Comunicación de contraseña temporal:

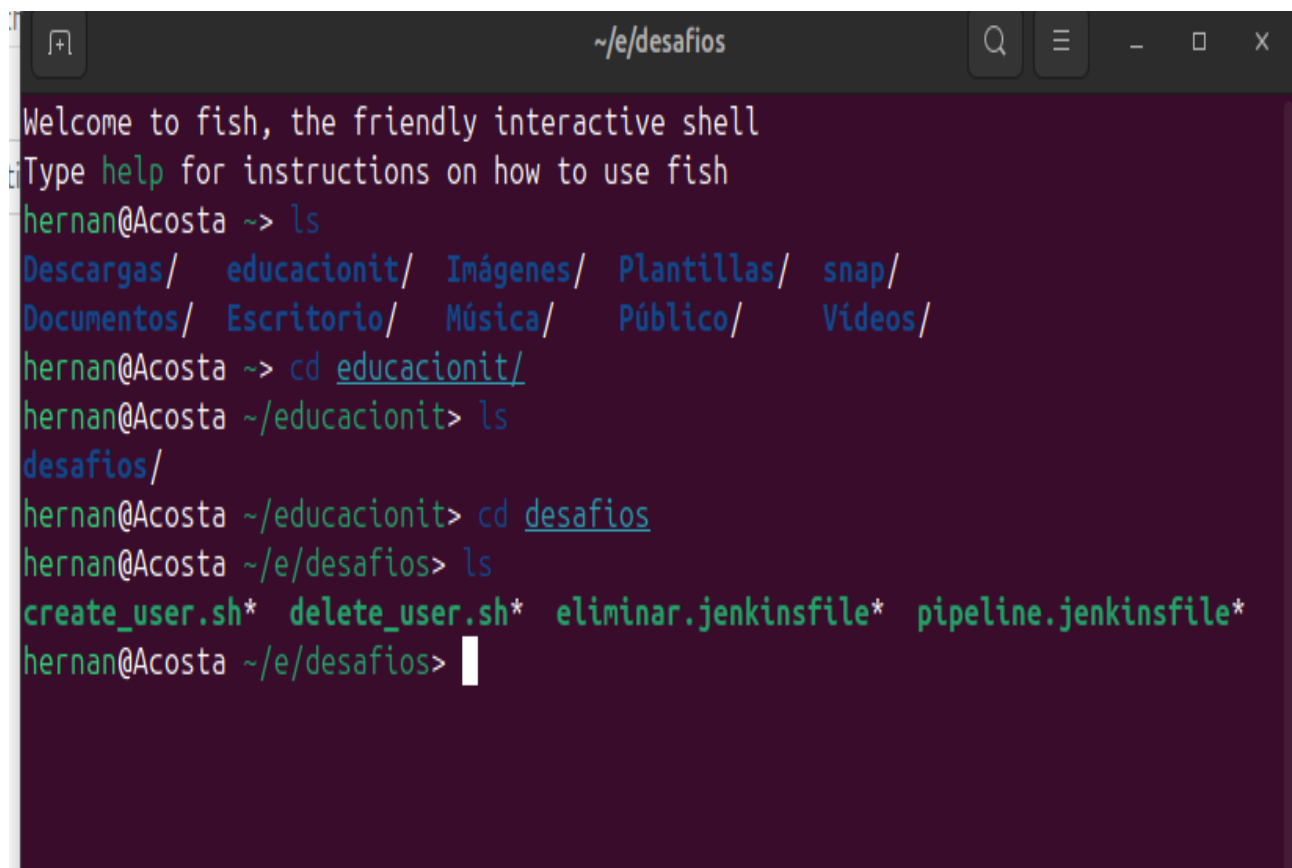
- **Integración:** se deben instalar Plug-in Email Extensión en Jenkins para enviar Email automáticamente cuando se creen los usuarios y las contraseñas.

#### PASOS PARA IMPLEMENTAR EL PIPELINE:

- **Configuración del repositorio de Github**
  - crear un repositorio en Github para contener los scripts de BASH y el JENKINSFILE.
- **Creación de scripts de bash:**
  - script para crear usuario: debe contener los datos de entrada (login, nombre, apellido, departamento), crear el usuario asignarle un grupo, crear un directorio home, y generar una contraseña temporal segura.
  - Script para eliminar usuarios: debe tomar el login del usuario y eliminar tanto el usuario como su grupo y directorio home.

#### Script para crear usuario

- Utilizamos un editor de texto EJ: nano para crear archivo (.sh), creando usuario con una contraseña temporal, o bien creamos el archivo con el comando touch create\_user.sh



```
~/e/desafios
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
hernan@Acosta ~-> ls
Descargas/  educacionit/  Imágenes/  Plantillas/  snap/
Documentos/ Escritorio/  Música/    Público/     Videos/
hernan@Acosta ~-> cd educacionit/
hernan@Acosta ~/educacionit> ls
desafios/
hernan@Acosta ~/educacionit> cd desafios
hernan@Acosta ~/e/desafios> ls
create_user.sh*  delete_user.sh*  eliminar.jenkinsfile*  pipeline.jenkinsfile*
hernan@Acosta ~/e/desafios>
```

```
hernan@Acosta ~/educacionit> ls
desafios/
hernan@Acosta ~/educacionit> cd desafios
hernan@Acosta ~/e/desafios> ls
create_user.sh* delete_user.sh* eliminar.jenkinsfile* pipeline.jenkinsfile*
hernan@Acosta ~/e/desafios> cat create_user.sh
#!/bin/bash
LOGIN=$1
FULL_NAME=$2
DEPARTMENT=$3
PASSWORD=$(openssl rand -base64 12)
sudo useradd -m -c "$FULL_NAME" -s /bin/bash -g "$DEPARTMENT" "$LOGIN"
echo "$LOGIN:$PASSWORD" | sudo chpasswd
sudo passwd -e "$LOGIN"
echo "Usuario: $LOGIN"
echo "Contraseña Temporal: $PASSWORD"
hernan@Acosta ~/e/desafios> cat delete_user.sh
```

```
hernan@Acosta ~/e/desafios> cat pipeline.jenkinsfile
pipeline {
  agent any
  parameters {
    string(name: 'LOGIN', description: 'Login del usuario (nombre.apellido)')
    string(name: 'FULL_NAME', description: 'Nombre y Apellido del usuario')
    choice(name:
      'DEPARTMENT',
      choices:
        'contabilidad\nfinanzas\ntecnología',
      description: 'Departamento del usuario')
  }
  stages {
    stage('Crear Usuario') {
      steps {
        sh '''
        bash create_user.sh "${LOGIN}" "${FULL_NAME}" "${DEPARTMENT}"
        '''
      }
    }
  }
}
```

hernan@Acosta ~/e/desafios>

#### 4. Script para Eliminar Usuarios

```
hernan@Acosta ~/e/desafios> cat delete_user.sh
#!/bin/bash
LOGIN=$1

sudo userdel -r "$FULL_NAME"

echo "usuario fue eliminado"

hernan@Acosta ~/e/desafios>
```

```
}
}
}
hernan@Acosta ~/e/desafios> ls
create_user.sh* delete_user.sh* eliminar.jenkinsfile* pipeline.jenkinsfile*
hernan@Acosta ~/e/desafios> cat eliminar.jenkinsfile
pipeline {
  agent any
  parameters {
    string(name: 'LOGIN', description: 'Login del usuario a eliminar (nombre.apellido)')
  }
  stages {
    stage('Eliminar Usuario') {
      steps {
        sh '''
bash delete_user.sh "${LOGIN}" "${FULL_NAME}" "${DEPARTMENT}"
'''
      }
    }
  }
}
```

«Captura desde 2024-11-21 17-15-50.png» selecciona

#### 5. Levantar los sripts creados a github

- una vez creados los scrip, lo subimos al repositorio en blacno que creamos en github para posteriormente clonarlo y actualizarlo.

##### Comados git utilizados:

- sudo git clone <url del repo> <clonamos el repositorio para poder subir los cambios con los script necesarios
- sudo add . <agregar los scrip>
- git status <verificar los cambios>
- sudo comint -a <agrega y confima todos los cambios en un solo pasa>

← → ↻ <https://github.com/hernan130/desafios> 🔍 ☆

hernan130 / **desafios** 🔍 Type ↗ to search + ↕ ↻ 📧 🌐

<> Code Issues Pull requests Actions Projects Security Insights Settings


**desafios** Private Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags Go to file Add file Code

**hernan130** modificados: delete\_user.sh e9c58f0 · 21 minutes ago 33 Commits

create_user.sh	modificados: create_user.sh	4 hours ago
delete_user.sh	modificados: delete_user.sh	21 minutes ago
eliminarjenkinsfile	modificados: delete_user.sh	21 minutes ago
pipeline.jenkinsfile	modificados: create_user.sh	4 hours ago

**README**

  
**Add a README**

**About** ⚙️  
No description, website, or topics provided.  
Activity  
0 stars  
1 watching  
0 forks

**Releases**  
No releases published  
[Create a new release](#)

**Packages**  
No packages published  
[Publish your first package](#)

← → ↻ <https://github.com/hernan130/desafios> 🔍 ☆

hernan130 / **desafios** 🔍 Type ↗ to search + ↕ ↻ 📧 🌐

<> Code Issues Pull requests Actions Projects Security Insights Settings

main **desafios / create\_user.sh** 🔍 Go to file ...

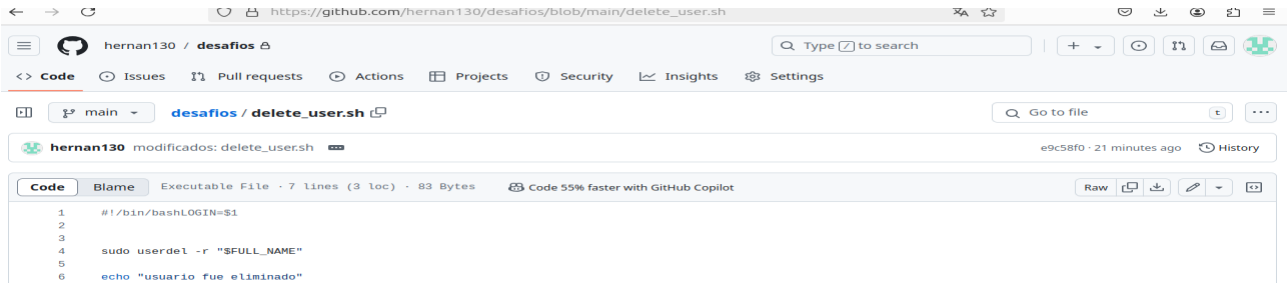
**hernan130** modificados: create\_user.sh e163375 · 4 hours ago History

**Code** Blame Executable File · 9 lines (9 loc) · 280 Bytes Code 55% faster with GitHub Copilot Raw 📄 ⬇️ ✎ ⌵

```
1 #!/bin/bash LOGIN=$1
2 FULL_NAME=$2
3 DEPARTMENT=$3
4 PASSWORD=$(openssl rand -base64 12)
5 sudo useradd -m -c "$FULL_NAME" -s /bin/bash -g "$DEPARTMENT" "$LOGIN"
6 echo "$LOGIN:$PASSWORD" | sudo chpasswd
7 sudo passwd -e "$LOGIN"
8 echo "Usuario: $LOGIN"
9 echo "Contraseña Temporal: $PASSWORD"
```

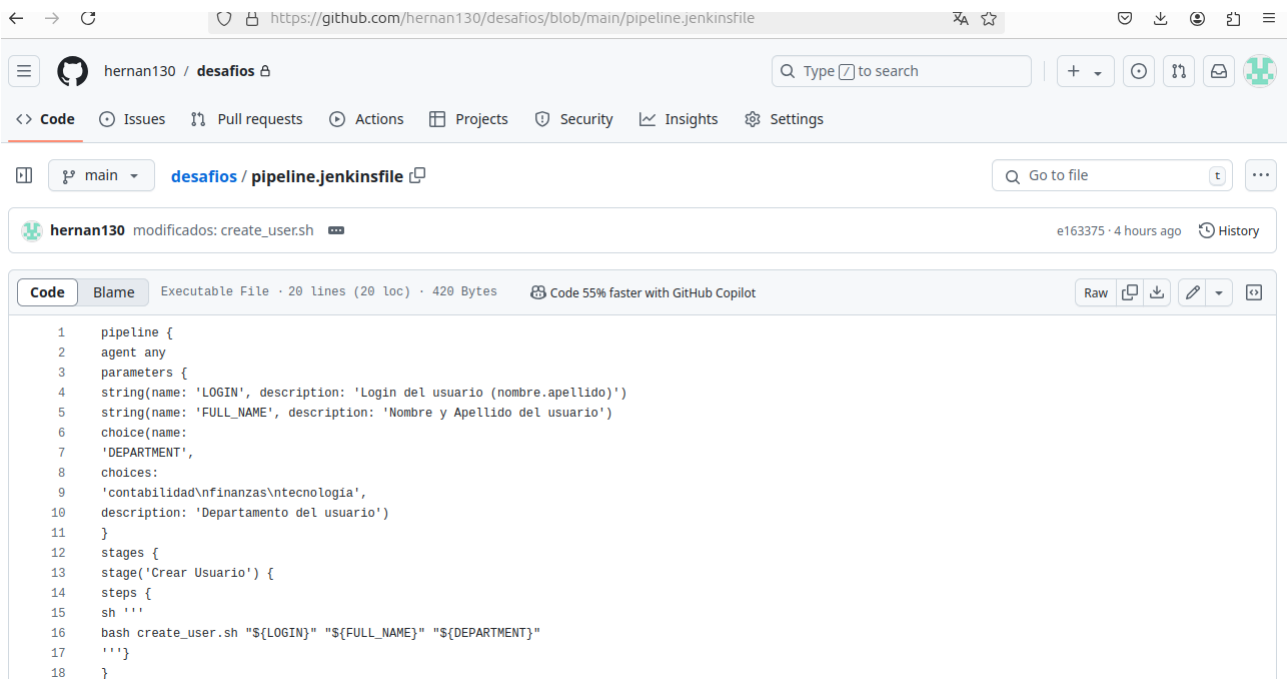
## 6. Configuración de Jenkins

Instalar y configurar Jenkins en un servidor que pueda comunicarse con las maquinas de Linux.



The screenshot shows a GitHub repository page for 'desafios' by user 'hernan130'. The file 'delete\_user.sh' is selected, showing its content. The script is a shell script that sets a login variable, prompts for a full name, and then deletes a user. It includes a confirmation message.

```
1 #!/bin/bash LOGIN=$1
2
3
4 sudo userdel -r "$FULL_NAME"
5
6 echo "usuario fue eliminado"
```



The screenshot shows a GitHub repository page for 'desafios' by user 'hernan130'. The file 'pipeline.jenkinsfile' is selected, showing its content. The file is a Jenkins pipeline script that defines parameters for user creation, including login, full name, and department. It includes a stage for creating a user, which calls a shell script 'create\_user.sh'.

```
1 pipeline {
2   agent any
3   parameters {
4     string(name: 'LOGIN', description: 'Login del usuario (nombre.apellido)')
5     string(name: 'FULL_NAME', description: 'Nombre y Apellido del usuario')
6     choice(name:
7       'DEPARTMENT',
8       choices:
9         'contabilidad\nfinanzas\ntecnologia',
10      description: 'Departamento del usuario')
11   }
12   stages {
13     stage('Crear Usuario') {
14       steps {
15         sh '''
16         bash create_user.sh "${LOGIN}" "${FULL_NAME}" "${DEPARTMENT}"
17         '''
18       }
19     }
20   }
21 }
```

The screenshot shows a GitHub repository page for 'desafios' by user 'hernan130'. The file 'eliminar.jenkinsfile' is selected, showing a Jenkins pipeline configuration. The pipeline has an agent 'any' and a parameter 'LOGIN' with a description 'Login del usuario a eliminar (nombre.apellido)'. It contains a stage 'Eliminar Usuario' with a step 'delete\_user.sh' that runs a bash script. The script takes 'LOGIN', 'FULL\_NAME', and 'DEPARTMENT' as arguments. The file is 15 lines long and 247 bytes.

```
1 pipeline {
2   agent any
3   parameters {
4     string(name: 'LOGIN', description: 'Login del usuario a eliminar (nombre.apellido)')
5   }
6   stages {
7     stage('Eliminar Usuario') {
8       steps {
9         sh '''
10        bash delete_user.sh "${LOGIN}" "${FULL_NAME}" "${DEPARTMENT}"
11        '''
12      }
13    }
14  }
15 }
```

1. Configurar el Plug-in de Jenkins para conectar el repositorio.
2. Configurar los jobs en Jenkins utilizando un (jenkinsfile) que definirá el pipeline declarativos incluye una primera etapas de creación de usuario y luego eliminación de usuarios.

The screenshot shows the Jenkins Configuration page for a pipeline named 'usuario\_jenkinsfile'. The 'SCM' is set to 'Git'. The 'Repository URL' is 'https://github.com/hernan130/desafio1.git'. The 'Credentials' field is filled with 'hacosta057@gmail.com/\*\*\*\*\* (tokendesafio\_1)'. There are buttons for '+ Add', 'Avanzado', and 'Add Repository'.

Panel de Control > develop-pipeline > usuario\_jenkinsfile > Configuration

### Configure

- General
- Advanced Project Options
- Pipeline

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/hernan130/desafio1.git

Credentials ?

hacosta057@gmail.com/\*\*\*\*\* (tokendesafio\_1)

+ Add

Avanzado ▾

Add Repository

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/hernan130/desafios.git

Credentials ?

hacosta057@gmail.com/\*\*\*\*\* (tokendesafio\_1)


+ Add

Guardar


Apply


## 7. Pruebas y Validación:


← → ↺ https://0127-190-7-36-128.ngrok-free.app/job/develop-pipeline/ ☆ 🔔 ⬇️ 🗨️ 📄 ☰


 **Jenkins** 🔍 búsqueda (CTRL+K) 🛡️ 1 👤 acosta hernan andres ▾ ➡️ Desconectar


Panel de Control > develop-pipeline >


 Status


 Configure


 + New Item


 Delete Folder


 Historial de trabajos


 Relacion entre proyectos

 Comprobar firma de archivos

 Rename

 Credentials

 **develop-pipeline**

 añadir descripción

All +

S	W	Nombre	Último Éxito	Último Fallo	Última Duración	
✓	☀️	delete_user1	1 Min 56 Seg #3	N/D	3,2 Seg	▶️
✓	☁️	desafio1	41 Min #3	51 Min #1	3,9 Seg	▶️

Icono: S M L ...

Trabajos en la cola ^

Estado del ejecutor de construcciones ^



desafios/pipeline.jenkinsl x Setup - ngrok x Contraseñas x develop-pipeline » desafio x +

← → ↻ <https://0127-190-7-36-128.ngrok-free.app/job/develop-pipeline/job/desafio1/2/console> 🔍 🌟 📄 ⬇️ 👤 📁 ☰

**Jenkins** 🔍 búsqueda (CTRL+K) 🛡️ 1 👤 acosta hernan andres 🚪 Desconectar

Panel de Control > develop-pipeline > desafio1 > #2

Status

## Salida de consola

Download

Copy

View as plain text

Changes

Console Output

Edit Build Information

Delete build '#2'

Timings

Git Build Data

Pipeline Overview

Pipeline Console

Restart from Stage

Replay

Pipeline Steps

Workspaces

```
16:09:08 Lanzada por el usuario acosta hernan andres
16:09:09 Obtained pipeline.jenkinsfile from git https://github.com/hernan130/desafios.git
16:09:09 [Pipeline] Start of Pipeline
16:09:09 [Pipeline] node
16:09:09 Running on Jenkins in /var/lib/jenkins/workspace/develop-pipeline/desafio1
16:09:09 [Pipeline] {
16:09:09 [Pipeline] stage
16:09:09 [Pipeline] { (Declarative: Checkout SCM)
16:09:09 [Pipeline] checkout
16:09:09 Selected Git installation does not exist. Using Default
16:09:09 The recommended git tool is: NONE
16:09:09 using credential tokendesafio_1
16:09:09 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/develop-pipeline/desafio1/.git
# timeout=10
16:09:09 Fetching changes from the remote Git repository
16:09:09 > git config remote.origin.url https://github.com/hernan130/desafios.git # timeout=10
16:09:09 Fetching upstream changes from https://github.com/hernan130/desafios.git
16:09:09 > git --version # timeout=10
16:09:09 > git --version # 'git version 2.43.0'
```

desafios/pipeline.jenkinsl x Setup - ngrok x Contraseñas x develop-pipeline » desafio x +

← → ↻ <https://0127-190-7-36-128.ngrok-free.app/job/develop-pipeline/job/desafio1/2/console> 🔍 🌟 📄 ⬇️ 👤 📁 ☰

Panel de Control > develop-pipeline > desafio1 > #2

Timestamps

[View as plain text](#)

☒ System clock time

☒ Use browser timezone

☐ Elapsed time

☐ None

```
16:09:11 [Pipeline] sh
16:09:11 + bash create_user.sh
16:09:11 sudo: a terminal is required to read the password; either use the -S option to read from
standard input or configure an askpass helper
16:09:11 sudo: se requiere una contraseña
16:09:11 sudo: a terminal is required to read the password; either use the -S option to read from
standard input or configure an askpass helper
16:09:11 sudo: se requiere una contraseña
16:09:11 sudo: a terminal is required to read the password; either use the -S option to read from
standard input or configure an askpass helper
16:09:11 sudo: se requiere una contraseña
16:09:11 Usuario:
16:09:11 Contraseña Temporal: lCp3LrXM6wFaEVia
16:09:11 [Pipeline] }
16:09:11 [Pipeline] // stage
16:09:11 [Pipeline] }
16:09:11 [Pipeline] // withEnv
16:09:11 [Pipeline] }
16:09:11 [Pipeline] // node
16:09:11 [Pipeline] End of Pipeline
16:09:11 Finished: SUCCESS
```

Status

Changes

Console Output

Edit Build Information

Delete build '#5'

Parámetros

Timings

Git Build Data

Pipeline Overview

Pipeline Console

Restart from Stage

Replay

Pipeline Steps

Workspaces

Previous Build

Timestamps

[View as plain text](#)

- ☒ System clock time  
☒ Use browser timezone  
☐ Elapsed time  
☐ None

## Salida de consola

```
16:41:57 Lanzada por el usuario acosta hernan andres
16:41:58 Obtained eliminar.jenkinsfile from git https://github.com/hernan130/desafios.git
16:41:58 [Pipeline] Start of Pipeline
16:41:59 [Pipeline] node
16:41:59 Running on Jenkins in /var/lib/jenkins/workspace/develop-pipeline/deleuser_d1
16:41:59 [Pipeline] {
16:41:59 [Pipeline] stage
16:41:59 [Pipeline] { (Declarative: Checkout SCM)
16:41:59 [Pipeline] checkout
16:41:59 Selected Git installation does not exist. Using Default
16:41:59 The recommended git tool is: NONE
16:41:59 using credential tokendesafio_1
16:41:59 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/develop-pipeline/deleuser_d1/.git # timeout=10
16:41:59 Fetching changes from the remote Git repository
16:41:59 > git config remote.origin.url https://github.com/hernan130/desafios.git # timeout=10
16:41:59 Fetching upstream changes from https://github.com/hernan130/desafios.git
16:41:59 > git --version # timeout=10
16:41:59 > git --version # 'git version 2.43.0'
16:41:59 using GIT_ASKPASS to set credentials tokendesafio_1
16:41:59 > git fetch --tags --force --progress -- https://github.com/hernan130/desafios.git +refs/heads/*:refs/remotes/origin/* # timeout=10
16:42:00 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
16:42:00 Checking out Revision e9c58f8dc87d8ddc836930737f428f5b54382ac3 (refs/remotes/origin/main)
16:42:00 > git config core.sparsecheckout # timeout=10
16:42:00 > git checkout -f e9c58f8dc87d8ddc836930737f428f5b54382ac3 # timeout=10
16:42:00 Commit message: "      modificados:   delete_user.sh      modificados:   eliminar.jenkinsfile"
16:42:00 > git rev-list --no-walk 46c2091a56ec76cafafa59cb02dbbf6cc25e3427 # timeout=10
16:42:00 [Pipeline] }
16:42:00 [Pipeline] // stage
16:42:00 [Pipeline] withEnv
16:42:00 [Pipeline] {
16:42:00 [Pipeline] stage
16:42:00 [Pipeline] { (Eliminar Usuario)
16:42:00 [Pipeline] sh
16:42:00 + bash delete_user.sh
16:42:00 sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
16:42:00 sudo: se requiere una contraseña
16:42:00 usuario fue eliminado
16:42:00 [Pipeline] }
16:42:00 [Pipeline] // stage
16:42:00 [Pipeline] }
16:42:00 [Pipeline] // withEnv
16:42:00 [Pipeline] }
16:42:00 [Pipeline] // node
16:42:00 [Pipeline] End of Pipeline
16:42:01 Finished: SUCCESS
```

Timestamps

[View as plain text](#) ▾

- ☐ System clock time
- ☒ Use browser timezone
- ☐ Elapsed time
- ☐ None

```
16:42:00 > git config core.sparsecheckout # timeout=10
16:42:00 > git checkout -f e9c58f0dc87d0ddc836930737f428f5b54382ac3 # ti
16:42:00 Commit message: "      modificados:      delete_user.sh      mo
16:42:00 > git rev-list --no-walk 46c2091a56ec76cafafa59cb02dbbf6cc25e34
16:42:00 [Pipeline] }
16:42:00 [Pipeline] // stage
16:42:00 [Pipeline] withEnv
16:42:00 [Pipeline] {
16:42:00 [Pipeline] stage
16:42:00 [Pipeline] { (Eliminar Usuario)
16:42:00 [Pipeline] sh
16:42:00 + bash delete_user.sh
16:42:00 sudo: a terminal is required to read the password; either use th
configure an askpass helper
16:42:00 sudo: se requiere una contraseña
16:42:00 usuario fue eliminado
16:42:00 [Pipeline] }
16:42:00 [Pipeline] // stage
16:42:00 [Pipeline] }
16:42:00 [Pipeline] // withEnv
16:42:00 [Pipeline] }
16:42:00 [Pipeline] // node
16:42:00 [Pipeline] End of Pipeline
16:42:01 Finished: SUCCESS
```