

JENKINS

Cristóbal Suárez Abad
OPTATIVA - 2º ASIR

Contenido

02. Curso de Jenkins - ¿Qué es Jenkins?.....	2
03. Curso de Jenkins – Instalación	4
04. Curso de Jenkins - Administrar Jenkins	8
05. Curso de Jenkins - Usuarios y Permisos en Jenkins.....	11
06. Curso de Jenkins - Job Free Style.....	14
07. Curso de Jenkins - Vistas y Folders.....	20
08. Curso de Jenkins - Variables de entorno	25
09. Curso de Jenkins - Introducción a Pipelines	30
10. Curso de Jenkins - Parametros en los Pipelines	39
11. Curso de Jenkins - Creación y uso de credenciales	43
12. Curso de Jenkins - Clonando un Repositorio.....	50
13. Curso de Jenkins - Trigger Job	52
14. Curso de Jenkins - JenkinsFile y Repositorios Git.....	56

02. Curso de Jenkins - ¿Qué es Jenkins?

A. Open-Source (Código Abierto)

Open-source o código abierto se refiere a un software cuyo código fuente está disponible públicamente y puede ser inspeccionado, modificado y distribuido por cualquiera, bajo una licencia específica.

Principio clave: Transparencia y colaboración comunitaria en el desarrollo.

B. Escalado Vertical (Scale-Up)

El **escalado vertical** (o *scale-up*) consiste en **aumentar la capacidad de un único servidor** o máquina.

- **¿Cómo?** Se añaden más recursos como **CPU, RAM o almacenamiento** a la máquina existente.
- **Ventaja:** Simple de implementar y gestionar.
- **Desventaja:** Limitado por el hardware físico máximo de la máquina y crea un único punto de fallo.

C. Escalado Horizontal (Scale-Out)

El **escalado horizontal** (o *scale-out*) consiste en **distribuir la carga de trabajo entre múltiples servidores** o máquinas.

- **¿Cómo?** Se añaden **más máquinas** (nodos) al sistema o clúster para que trabajen en paralelo.
- **Ventaja:** Alta disponibilidad (si una máquina falla, las otras continúan operando), escalabilidad casi ilimitada y mejor elasticidad.
- **Desventaja:** Mayor complejidad en la administración, balanceo de carga y en el diseño de la aplicación.

D. On-Premise (Local)

On-premise se refiere al modelo de implementación donde una organización **compra, instala y ejecuta el software y su infraestructura tecnológica** (servidores, redes, almacenamiento) **dentro de sus propias instalaciones físicas**.

- **Principio clave:** La organización tiene el **control total** sobre la infraestructura, pero también es responsable de su mantenimiento, seguridad y gestión.
- **Contraste:** Se opone al modelo de **Nube (Cloud)**, donde la infraestructura es gestionada por un proveedor externo.

Jenkins es un Software Open Source de automatización, generalmente utilizado para compilar, probar y desplegar proyectos de software de manera continua.

El probablemente el software de automatización más grande y popular de todos.

Está escrito en Java. Es extensible. Permite escalamiento vertical y horizontal. Es ON-Premise. Pipeline As Code.

03. Curso de Jenkins – Instalación

Tiene formas variadas de instalación. Para este caso usamos Docker.

```
docker run -d -p 8080:8080 -p 50000:50000 -v jenkins_home:/var/jenkins_home  
jenkins/jenkins:lts-jdk17
```

```
C:\Users\Cristobal>docker run -d -p 8080:8080 -p 50000:50000 -v jenkins_home:/var/jenkins_home jenkins/jenkins:lts-jdk17  
Unable to find image 'jenkins/jenkins:lts-jdk17' locally  
lts-jdk17: Pulling from jenkins/jenkins  
13cc39f8244a: Downloading [=====>  
] 31.46MB/49.29MB  
368476d37131: Download complete  
33300af18dd0: Download complete  
13cc39f8244a: Pull complete  
368476d37131: Pull complete  
33300af18dd0: Pull complete  
98ae2e79fe48: Pull complete  
a37b858bb47a: Pull complete  
a1168de58ed4: Pull complete  
e4beac64dfffa: Pull complete  
744b4792e083: Pull complete  
de160a647174: Pull complete  
dc2a77f462ea: Pull complete  
c27509c3e53b: Pull complete  
05a7d9a8b608: Pull complete  
Digest: sha256:ddcec7eb31abeb9aa864510a19874d03d53948b040e19461e3432be5455b1f  
Status: Downloaded newer image for jenkins/jenkins:lts-jdk17  
9eab68d6b058c0c4b6deae29162f6f14cbd1d09e248fbb2a25b0aacc508fe964
```

Ahora podemos acceder usando el navegador web:

<http://localhost:8080>

Nos indicará que debemos usar una contraseña generada dentro del contenedor.

```
docker exec -it "código contenedor" bash
```

```
docker exec -it
```

```
9eab68d6b058c0c4b6deae29162f6f14cbd1d09e248fbb2a25b0aacc508fe964 bash
```

```
cat /var/jenkins_home/secrets/initialAdminPassword
```

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/var/jenkins_home/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

```
.....
```

Continue

En este caso elegimos “Instalar los plugins sugeridos”. Puede tardar uno o dos minutos.

Getting Started

Bienvenido a Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✗ Credentials Binding
✗ Timestamper	✗ Workspace Cleanup	✗ Ant	✗ Gradle
✗ Pipeline	✗ GitHub Branch Source	✗ Pipeline: GitHub Groovy Libraries	✗ Pipeline Graph View
✗ Git	✗ SSH Build Agents	✗ Matrix Authorization Strategy	✗ LDAP
✗ Email Extension	✗ Mailer	✗ Dark Theme	

```

** commons-lang3 v3.x Jenkins API
** Ionicons API
Folders
OWASP Markup Formatter
** ASM API
** JSON Path API
** Structs
** Pipeline: Step API
** commons-text API
** Token Macro
Build Timeout
** bouncycastle API
** Credentials
** Plain Credentials
** - required dependency

```

Jenkins 2.528.2

Ahora creamos el primer usuario. Yo creo el “admin”. Tendrás permisos máximos.

Getting Started

Create First Admin User

Usuario

Contraseña

Confirma la contraseña

Nombre completo

Jenkins 2.528.2

Skip and continue as admin

Save and Continue

Esto se puede modificar: dominio, ip, etc.

Getting Started

Instance Configuration

Jenkins URL:

`http://localhost:8080/`

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.528.2

Not now

Save and Finish

Ese era el último paso.

The screenshot shows the Jenkins home page at `http://localhost:8080`. At the top, there's a navigation bar with back, forward, and refresh icons. Below it, the Jenkins logo is displayed next to the word "Jenkins". There are two main buttons: "+ Nueva Tarea" and "Historial de trabajos". To the right, a large banner says "¡Bienvenido a Jenkins!" and provides instructions: "This page is where your Jenkins jobs will be displayed or start building a software project." Below this, a button says "Start building your software project". On the left, a section titled "Trabajos en la cola" shows a dropdown menu with the message "No hay trabajos en la cola". At the bottom, a progress bar for "Estado del ejecutor de construcciones" shows "0/2". A "Create a job" button is located in the bottom right corner.

04. Curso de Jenkins - Administrar Jenkins

Esquina derecha: Administrar Jenkins. Solo deben poder acceder usuarios con permisos elevados.

The screenshot shows the Jenkins System Configuration page. At the top right, there is a navigation bar with a magnifying glass icon, a gear icon (highlighted with a red dot), and a user profile icon. A tooltip box labeled "Administrar Jenkins" with the subtext "One or more notifications" is overlaid on the gear icon. Below the navigation bar, the page title "System Configuration" is visible. The main content area lists several configuration items, each with an icon and a brief description:

- System**: Configurar variables globales y rutas.
- Tools**: Configure tools, their locations and automatic installers.
- Plugins**: Añadir, borrar, desactivar y activar plugins que extienden la funcionalidad de Jenkins.
- Nodes**: Añadir, borrar, gestionar y monitorizar los nodos sobre los que Jenkins ejecuta tareas.
- Clouds**: Add, remove, and configure cloud instances to provision agents on-demand.
- Appearance**: Configure the look and feel of Jenkins

Security



Security

Seguridad en Jenkins. Define quién tiene acceso al sistema (autenticación) y qué puede hacer (autorización)



Credentials

Configure credentials



Credential Providers

Configure the credential providers and types



Users

Crear/borrar/editar usuarios que puedan utilizar Jenkins

Status Information

Status Information



Información del sistema

Muestra información del entorno que puedan ayudar a la solución de problemas.



System Log

El log del sistema captura la salida de la clase `java.util.logging` en todo lo relacionado con Jenkins.



Estadísticas de Carga

Comprobar la utilización de los recursos y comprobar si es necesario añadir nuevos nodos para la ejecución de tareas.



Acerca de Jenkins

Eche un vistazo a la información sobre la versión y la licencia.

Troubleshooting



Datos antiguos

Scrub configuration files to remove remnants from old plugins and earlier versions.

Sistema: Toda la información: rutas, variables globales, etc. La cuenta de GitHub. Y otras cuentas.

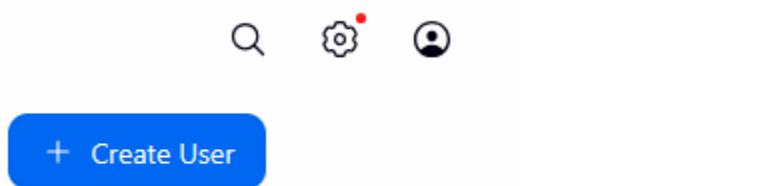
Herramientas: Jenkins puede usar todos los recursos del Sistema (servidor). Pero también se pueden usar las herramientas de los plugins (Plugins → Available Plugins).

Nodes y Clouds: Escalamiento Horizontal (Meter más equipos).

Credenciales: Configurar usuarios y sus credenciales.

05. Curso de Jenkins - Usuarios y Permisos en Jenkins

Administrar Jenkins → Users → Crear usuario.



que hayan hecho 'commits' a proyectos.

Crear un usuario

Usuario

test

Contraseña

Confirma la contraseña

Nombre completo

test

Dirección de email

jajaja@nolodoy.com

Crear un usuario

The screenshot shows a list of users under the heading "User ID ↓". There are two entries: "admin" and "test". Each entry consists of a small user icon followed by the username in blue text.

Ahora el configuramos los permisos:

Administrar Jenkins → Security.

- Authentication:

Authentication

Disable "Keep me signed in" ?

Seguridad

Usar base de datos de Jenkins

Delegar seguridad al contenedor de servlets

LDAP

Usar base de datos de Jenkins

None

Usuarios autenticados tienen privilegios para todo

En Autorización debemos poner “Configuración de Seguridad”. Eso nos permitirá establecer permisos por usuarios y grupos.

Autorización

The screenshot shows the 'Authorization' section with the following configuration:

- Usuarios autenticados tienen privilegios para todo** (Authenticated users have privileges for everything)
- Configuración de seguridad** (Security configuration)
 - Cualquiera puede hacer cualquier acción (Anybody can do anything)
 - Estrategia de seguridad para el proyecto (Project security strategy)
 - Modo 'legacy' (Legacy mode)
 - Usuarios autenticados tienen privilegios para todo** (Authenticated users have privileges for everything)

En la imagen de abajo se ve configuración máxima para “admin” y el resto de usuarios autenticados, podrán leer las configuraciones.

Autorización

The screenshot shows the 'Authorization' configuration interface with the following settings:

Usuario/Grupo	Global		
	Administer	Read	Create
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authenticated Users	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
admin	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

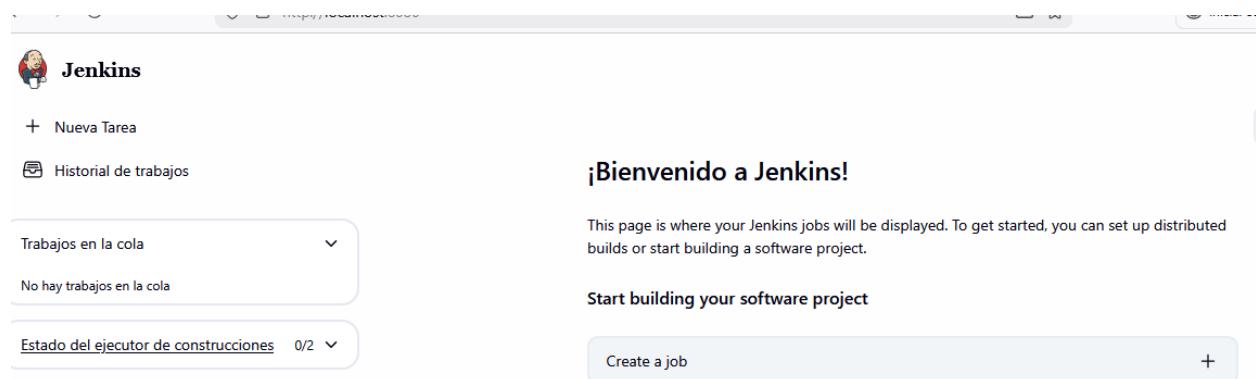
Buttons at the bottom include: Add user, Add group, Guardar (Save), and Apply.

06. Curso de Jenkins - Job Free Style

Los Jobs es la columna vertebral de Jenkins.

Panel de Control → Nueva tarea

O Create a Job



The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with a Jenkins logo, a 'Nueva Tarea' button, and a 'Historial de trabajos' link. Below the sidebar, there's a dropdown menu set to 'Trabajos en la cola' which shows 'No hay trabajos en la cola'. To the right, a main panel displays the message '¡Bienvenido a Jenkins!' and instructions: 'This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.' Below this, a 'Start building your software project' button is visible. At the bottom right of the main panel is a 'Create a job' button with a '+' sign next to it.

Elegimos nombre y le damos a “Crear un Proyecto de Estilo libre” para este caso.

Nuevo Tarea

Nuevo Tarea

Enter an item name

Prueba01

Select an item type



Crear un proyecto de estilo libre

Classic, general-purpose job type that checks out from up to one !
steps like archiving artifacts and sending email notifications.

Una vez creado, en la página principal veremos esto:

The screenshot shows the Jenkins dashboard with a single job listed. The job is named "Prueba01". It has a status icon (blue circle with a white dot) and a build status icon (yellow sun). The last build was successful ("Último Éxito: N/D") and the last failure was also "N/D". The last duration is also "N/D". There is a green "View" button to the right.

S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración
		Prueba01	N/D	N/D	

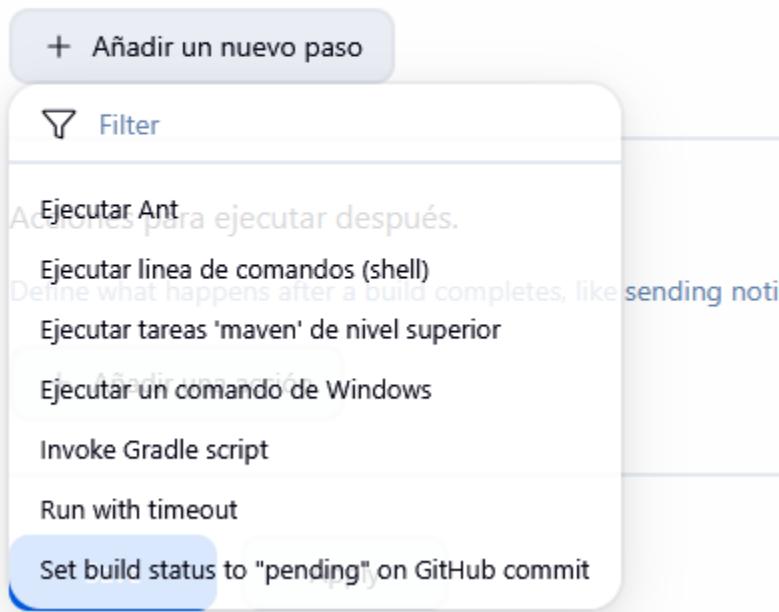
Y también podremos configurar el JOB

The screenshot shows the Jenkins configuration page for the job "Prueba01". The left sidebar lists configuration sections: General, Triggers, Environment, Build Steps, and Post-build Actions. The "General" section is selected and active. On the right, there are fields for "Descripción" (Description) and "Plain text" (Visualizar) for the build. Under "Post-build Actions", three checkboxes are shown: "Desechar ejecuciones antiguas" (Delete old builds), "Esta ejecución debe parametrizarse" (This build must be parameterized), and "GitHub project" (GitHub project).

Nos vamos a “Build Steps” y le damos a “Ejecutar linea de commandos (Shell)”.

Build Steps

Automate your build process with ordered tasks like code com



Escribimos los commandos

The screenshot shows the configuration for the 'Ejecutar linea de comandos (shell)' step. It includes a section for 'Comando' (Command) with a link to 'Visualizar la lista de variables de entorno disponibles' (View available environment variable list). The command text area contains the following code:

```
date  
pwd  
echo "Hola mundo" >> saludos.txt
```

Guardamos y le damos a “Construir Ahora”.



Jenkins / Prueba01

- [Estado Actual](#)
- [Cambios](#)
- [Zona de Trabajo](#)
- [Construir ahora](#)

En el panel general Podemos ver si ha sido exitoso o no. Le damos al número que indica la “vez” que ha sido hecho y le damos a “Console output”

S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración
		Prueba01	9.8 Seg #1	N/D	0.31 Seg

</> Cambios
Console Output

Nos mostrará el resultado de la ejecución de los commandos del JOB.

Salida de consola

```

Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/Prueba01
[Prueba01] $ /bin/sh -xe /tmp/jenkins628162446541213475.sh
+ date
Sun Dec 7 09:57:27 UTC 2025
+ pwd
/var/jenkins_home/workspace/Prueba01
+ echo Hola mundo
+ touch hola_caracola.txt
+ ls -l
total 4
-rw-r--r-- 1 jenkins jenkins 0 Dec 7 09:57 hola_caracola.txt
-rw-r--r-- 1 jenkins jenkins 22 Dec 7 09:57 saludos.txt
Finished: SUCCESS

```

también podemos meternos desde “Historia de tareas”.

Como has visto antes, hemos creado dos archivos. Para poder visualizarlos nos vamos al JOB → Zona de trabajo.

Workspace of Prueba01 on principal

- Prueba01 /
- hola_caracola.txt 7 dic 2025 9:57:27 0 B
- saludos.txt 7 dic 2025 9:57:27 22 B

(Descargar archivos en formato zip)

Hola mundo
Hola mundo

También puedes meter archivos desde la terminal (dentro del contenedor).

```
jenkins@9eab68d6b058:~/workspace/Prueba01$ pwd
/var/jenkins_home/workspace/Prueba01
jenkins@9eab68d6b058:~/workspace/Prueba01$ echo "hackeado" >> has_sido_hackeado.txt
jenkins@9eab68d6b058:~/workspace/Prueba01$ cat has_sido_hackeado.txt
hackeado
jenkins@9eab68d6b058:~/workspace/Prueba01$ |
```

Y en el resultado veremos que está ahí.

```
total 8
-rw-r--r-- 1 jenkins jenkins 9 Dec 7 10:02 has_sido_hackeado.txt
-rw-r--r-- 1 jenkins jenkins 0 Dec 7 10:04 hola_caracola.txt
-rw-r--r-- 1 jenkins jenkins 44 Dec 7 10:04 saludos.txt
+ cat has_sido_hackeado.txt
hackeado
Finished: SUCCESS
```

ATENCIÓN: estos archivos se crean con el usuario Jenkins. A tener en cuenta.

Cuando falla, sale esto:

S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración	
		Prueba01	2 Min 54 Seg #4	7.1 Seg #5	22 Ms	


```
+ cat asdasdasd.txt
cat: asdasdasd.txt: No such file or directory
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

07. Curso de Jenkins - Vistas y Folders

Sirve para organizar. En Panel Principal.

The screenshot shows the Jenkins main interface with a sidebar on the left containing icons for 'Nueva Tarea' (New Item) and 'Historial de trabajos' (Job History). On the right, there are two buttons: 'Nueva vista' (New View) and 'Todo' (All). A modal dialog titled 'Nueva vista' is open in the center. It has a 'Nombre de vista' (View Name) input field containing 'Free Style Projects'. Under the 'Type' section, 'Lista de vistas' (List View) is selected, indicated by a blue dot next to the radio button. A description below it states: 'Mostrar proyectos como una lista simple. Puedes elegir qué proyectos mostrar en cada vista.' (Show projects as a simple list. You can choose which projects to show in each view.) The 'Mi vista' (My View) option is also listed with its description: 'Esta vista muestra automáticamente todas las tareas a las que el usuario puede acceder.' (This view automatically shows all tasks to which the user has access.) At the bottom of the dialog is a blue 'Create' button.

Editar la vista

Nombre

Free Style Projects

Descripción

Describe the purpose of this view.

Estos son los proyectos Free Style Jobs

Añadimos Proyecto.

Proyectos

Select the jobs to show in this view.



Recurse in subfolders



Prueba01



Utilizar una expresión regular para incluir proyectos dentro de la vista ?

Ahora en el Panel Principal veremos que Podemos elegir entre el recién creado o Todos.

The screenshot shows the Jenkins main interface. On the left, there's a sidebar with icons for 'Nueva Tarea' (New Item), 'Historial de trabajos' (Job History), 'Editar vista' (Edit View), and 'Borrar vista' (Delete View). Below this is a dropdown menu labeled 'Trabajos en la cola' (Jobs in Queue). On the right, a list titled 'Estos son los proyectos Free Style Jobs' (These are the Free Style Jobs) shows one item: 'Prueba01' with a status icon (red circle with a white 'X') and a timestamp '21 M'. A navigation bar at the top right includes tabs for 'Free Style Projects', 'Todo', and a '+' button.

S	W	Nombre ↓	Último
		Prueba01	21 M

Para agregar un nuevo Job en esta vista, me ubico en la lista y le doy a nuevo JOB o tarea.

This screenshot shows the same Jenkins interface as above, but now with two items in the list: 'Prueba01' and 'Prueba02'. Both have green checkmark icons next to them. The table columns are: S (Status), W (Weather), Nombre ↓ (Name), Último Éxito (Last Success), Último Fallo (Last Failure), and Última Duración (Last Duration).

S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración
		Prueba01	28 Min #4	25 Min #5	22 Ms
		Prueba02	5.4 Seg #1	N/D	91 Ms

Proyecto sin nada.

Proyectos de Pipeline

This screenshot shows the Jenkins interface with the 'Pipeline Projects' tab selected. The main area is empty, except for a large blue cloud icon with rain and a lightning bolt. Below it, the text 'Esta vista no tiene ningún proyecto asociado.' (This view does not have any associated projects.) is displayed, along with a blue 'Añadir.' (Add) button.

Otra manera de organizar es con Folder:

Nueva tarea → Folder:

entornos, ejecutar sobre plataformas concretas, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Despliegues / Configuration

General

Display Name ?

Despliegues

Description

Aquí estarán los despliegues con Free Style Jobs

Free Style Projects	Pipeline Projects	Todo	+		
S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración
		Despliegues	N/D	N/D	N/D
		Prueba01	37 Min #4	34 Min #5	22 Ms
		Prueba02	9 Min 5 Seg #1	N/D	91 Ms

Dentro de la carpeta Podemos seguir creando vistas y jobs.

Despliegues

Aquí estarán los despliegues con Free Style Jobs

[Nueva vista](#)

All



This folder is empty

[Create a job](#)

Despliegues

Aquí estarán los despliegues con Free Style Jobs

[añadir descripción](#)

All



S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración
...	...	Despliegue-01	N/D	N/D	N/D
...	...	Despliegue-02	N/D	N/D	N/D

El concepto es el mismo.



Jenkins

/ Free Style Projects / Despliegues / Despliegue-01 / #1 / Salida de consola

Estatus

Cambios

Console Output

Editar información de la ejecución

Delete build '#1'

Timings

Salida de consola

```

Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/Despliegues/Despliegue-01
[Despliegue-01] $ /bin/sh -xe /tmp/jenkins5112744244863005054.sh
+ echo ¿En que carpeta estoy?
¿En que carpeta estoy?
+ pwd
/var/jenkins_home/workspace/Despliegues/Despliegue-01
Finished: SUCCESS

```

08. Curso de Jenkins - Variables de entorno

Listado de variables de entorno disponibles. Pueden aumentar según plugins instalados.

The following variables are available to shell and batch build steps:

- BRANCH_NAME**
For a multibranch project, this will be set to the name of the branch being built, for example in case you wish to deploy to primary branches; if corresponding to some kind of change request, the name is generally arbitrary (refer to CHANGE_ID and CHANGE_TITLE).
- BRANCH_IS_PRIMARY**
For a multibranch project, if the SCM source reports that the branch being built is a primary branch, this will be set to "true". Note that it is possible to have more than one branch as a primary branch while others may not supply this information.
- CHANGE_ID**
For a multibranch project corresponding to some kind of change request, this will be set to the change ID, such as a pull request identifier.
- CHANGE_URL**
For a multibranch project corresponding to some kind of change request, this will be set to the change URL, if supported; else to the URL of the latest commit.
- CHANGE_TITLE**
For a multibranch project corresponding to some kind of change request, this will be set to the change title, if supported; else to the title of the latest commit.

BUILD_NUMBER: The current build number, such as "153".

Ejecutar linea de comandos (shell) ?

Comando

[Visualizar la lista de variables de entorno disponibles](#)

```
echo $BUILD_NUMBER
echo ${BUILD_NUMBER}
```

✓ Salida de consola



```
Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/Despliegues/Despliegue-02
[Despliegue-02] $ /bin/sh -xe /tmp/jenkins12842009565283426273.sh
+ echo 2
2
+ echo 2
2
Finished: SUCCESS
```

Hay variables de entorno cuyo valor es nulo. Se debe al contexto.

≡ Ejecutar linea de comandos (shell) ?

Comando

[Visualizar la lista de variables de entorno disponibles](#)

```
echo $BUILD_NUMBER
echo ${BUILD_NUMBER}
echo $BRANCH_NAME
```

No muestra nada:

```
Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/Despliegues/Despliegue-02
[Despliegue-02] $ /bin/sh -xe /tmp/jenkins5211315330723831990.sh
+ echo 3
3
+ echo 3
3
+ echo

Finished: SUCCESS
```

Podemos crear variables de entorno personalizadas:

Administrar Jenkins → System → Propiedades Globales: Marca la casilla de “Variables de entorno” → Añadir.

Propiedades globales

- Disable deferred wipeout on this node [?](#)
- Disk Space Monitoring Thresholds
- Localización de herramientas
- Variables de entorno

Lista de nombre-valores [?](#)

nombre

JAMON

valor

Está rico

[Save](#)

[Apply](#)

 Salida de consola Download

```
Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/Despliegues/Despliegue-02
[Despliegue-02] $ /bin/sh -xe /tmp/jenkins11923896969025189597.sh
+ echo 4
4
+ echo 4
4
+ echo

+ echo Está rico
Está rico
Finished: SUCCESS
```

Las variables de entorno (sus datos) son visibles. Así que no pongas datos sensibles (contraseñas, etc).

09. Curso de Jenkins - Introducción a Pipelines

Nos metemos en la vista de Pipeline Projects y creamos nueva tarea. Elegimos Pipeline.



Nuevo Tarea

Enter an item name

Primer Pipeline

Select an item type



Crear un proyecto de estilo libre

Classic, general-purpose job type that checks out from up to one step like archiving artifacts and sending email notifications.



Pipeline

Gestiona actividades de larga duración que pueden abarcar varios pipelines (conocidas anteriormente como workflows) y/o para la o articular fácilmente con tareas de tipo freestyle.



Crear un proyecto multi-configuration

Adecuado para proyectos que requieran un gran número de configuraciones, ejecutar sobre plataformas concretas, etc.

OK



Add to current view

Creates a container that stores nested items in it. Useful for grouping related items.

The screenshot shows the Jenkins Pipeline Configuration page for a project named 'Primer Pipeline'. The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area is titled 'Pipeline' with the sub-instruction 'Define your Pipeline using Groovy directly or pull it from source control.' Below this is a 'Definition' section labeled 'Pipeline script'. A code editor window displays the following Groovy script:

```
1 pipeline {  
2     agent any  
3  
4     stages {  
5         stage ("Saludo 1") {  
6             steps {  
7                 echo "Hola desde Saludo 1"  
8             }  
9         }  
10    }  
11 }
```

At the bottom of the editor are 'Save' and 'Apply' buttons.

```
pipeline {  
    agent any  
  
    stages {  
        stage ("Saludo 1") {  
            steps {  
                echo "Hola desde Saludo 1"  
            }  
        }  
    }  
}
```

Guardamos y construimos.

-

► Construir ahora

 Configurar

 Borrar Pipeline

 Move

 Stages

 Rename

 Pipeline Syntax

Builds

Today

... #1 11:21   

[http://localhost:8080/view/Pipeline Projects/job/Primer Pipeline/changes](http://localhost:8080/view/Pipeline%20Projects/job/Primer%20Pipeline/changes)

Usa después la vista de Pipeline Overview

← → ⌂ http://localhost:8080/view/Pipeline Projects/job/Primer Pipeline/1/pipeline-overview/

 Jenkins / Pipeline Projects / Primer Pipeline / #1 / Pipeline Overview

✓ #1

Manually run by admin Comenzó hace 1 Min 28 Seg Queued 3 Ms Took 9.9 Seg

Graph

```
graph LR; Start(( )) --> Saludo1[Saludo 1]; Saludo1 --> End(( ));
```

Start — Saludo 1 — End

Search

✓ Saludo 1 0,66s

✓ Saludo 1

✓ Hola desde Saludo 1

0 Hola desde Saludo 1

cript :

```
1 v pipeline {  
2     agent any  
3  
4 v     stages {  
5 v         stage ("Saludo 1") {  
6 v             steps {  
7                 echo "Hola desde Saludo 1"  
8             }  
9         }  
10 v        stage ("Saludo 2") {  
11 v            steps {  
12                 echo "Hola desde Saludo 2, paso 1"  
13                 echo "Hola desde Saludo 2, paso 2"  
14             }  
15         }
```

< #3

Manually run by admin Comenzó hace 14 Seg Queued 2 Ms Took 6.5 Seg

Graph



Q Search

Saludo 2

Saludo 1 0,67s

Saludo 2 0,96s

Hola desde Saludo 2, paso 1 >

Hola desde Saludo 2, paso 2 ▾

Podemos poner una variable de entorno en la Pipeline

```
pipeline {
    agent any
    environment{
        CSA_INSTAGRAM="@csa_insta"
    }
    stages{
        stage ("Saludo 1"){
            steps{
                echo "Hola desde Saludo 1"
            }
        }
        stage ("Saludo 2"){
            steps{
                echo "Hola desde Saludo 2, paso 1"
                echo "Hola desde Saludo 2, paso 2"
            }
        }
        stage ("Imprimir variable de entorno"){
            steps{
                echo "${env.CSA_INSTAGRAM}"
                echo "${CSA_INSTAGRAM}"
                echo "$CSA_INSTAGRAM"
            }
        }
    }
}
```

✓ Imprimir variable de entorno

✓ @csa_insta ▾

✗ @csa_insta

✓ @csa_insta ▾

✗ @csa_insta

✓ @csa_insta >

Hasta ahora hemos imprimido, para ejecutar comandos usamos “sh”.

```
sh "echo 'hola mundo'"  
}  
+ echo hola mundo  
hola mundo  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] }  
[Pipeline] // withEnv  
[Pipeline] }  
[Pipeline] // node  
[Pipeline] End of Pipeline  
Finished: SUCCESS
```

sh con variables.

```
sh "echo 'hola mundo'"  
sh "${env.CSA_INSTAGRAM}"  
sh "${CSA_INSTAGRAM}"  
sh "$CSA_INSTAGRAM"  
}  
  
+ echo hola mundo  
hola mundo  
[Pipeline] sh  
+ echo @csa_insta  
@csa_insta  
[Pipeline] sh  
+ echo @csa_insta  
@csa_insta  
[Pipeline] sh  
+ echo @csa_insta  
@csa_insta  
[Pipeline] }
```

```
pipeline {
    agent any
    environment {
        CSA_INSTAGRAM="@csa_insta"
    }
    stages {
        stage ("Saludo 1") {
            steps {
                echo "Hola desde Saludo 1"
            }
        }
        stage ("Saludo 2") {
            steps {
                echo "Hola desde Saludo 2, paso 1"
                echo "Hola desde Saludo 2, paso 2"
            }
        }
        stage ("Imprimir variable de entorno") {
            steps {
                echo "${env.CSA_INSTAGRAM}"
                echo "${CSA_INSTAGRAM}"
                echo "$CSA_INSTAGRAM"

                sh "echo 'hola mundo'"
                sh "echo '${env.CSA_INSTAGRAM}'"
                sh "echo '${CSA_INSTAGRAM}'"
                sh "echo '$CSA_INSTAGRAM'"

            }
        }
        stage ("Imprimir variable de entorno definida en System") {
            steps {
                sh "echo '${env.JAMON}'"
                sh "echo '${JAMON}'"
                sh "echo '$JAMON'"

            }
        }
    }
}
```

10. Curso de Jenkins - Parámetros en los Pipelines

Para ver las variables de entorno: <http://localhost:8080/env-vars.html/>

Vamos a probar que una variable de entorno que hemos definido en el Pipeline, cambie de valor en el mismo documento.

Debe de estar dentro de script:

```
pipeline {
    agent any
    environment {
        CSA_INSTAGRAM="@csa_insta"
    }
    stages {
        stage ("Saludo 1") {
            steps {
                echo "Hola desde Saludo 1"
            }
        }
        stage ("Saludo 2") {
            steps {
                echo "Hola desde Saludo 2, paso 1"
                echo "Hola desde Saludo 2, paso 2"
            }
        }
        stage ("Imprimir variable de entorno") {
            steps {
                echo "${env.CSA_INSTAGRAM}"
                echo "${CSA_INSTAGRAM}"
                echo "$CSA_INSTAGRAM"

                sh "echo 'hola mundo'"
                sh "echo '${env.CSA_INSTAGRAM}'"
                sh "echo '${CSA_INSTAGRAM}'"
                sh "echo '$CSA_INSTAGRAM'"

            }
        }
        stage ("Imprimir variable de entorno definida en System") {
            steps {
                sh "echo '${env.JAMON}'"
                sh "echo '${JAMON}'"
            }
        }
    }
}
```

```
        sh "echo '$JAMON'"  
    }  
}  
stage ("Cambiar valor a variable de entorno definida previamente en documento  
Pipeline") {  
    steps {  
        script {  
            CSA_INSTAGRAM="@cristobal_insta"  
            echo "${CSA_INSTAGRAM}"  
        }  
    }  
}  
}
```

¿Cómo hacer para agregar un valor manualmente antes de que se ejecute el Pipeline?
Encima de Environment escribes parameters.

```
pipeline {  
    agent any  
    parameters {  
        string(name:'APELIDO', description:'Ingrese su Apellido')  
    }  
    environment {  
        CSA_INSTAGRAM="@csa_insta"  
    }  
}
```

Ahora nos saldrá: “Build with parameters”.

The screenshot shows the Jenkins Pipeline interface for a project named "Segundo Pipeline". On the left, there's a sidebar with options like Status, Changes, Build with Parameters, Configurar, Borrar Pipeline, Move, Stages, and Rename. The "Build with Parameters" option is selected. To its right, it says "Esta ejecución requiere parámetros adicionales:" followed by "APELLIDO" and a text input field containing "Ingrese su Apellido". Below this is a large input field with the value "Suárez". At the bottom right are two buttons: a green "Ejecución" button and a grey "Cancel" button. In the bottom right corner of the main area, there's a terminal-like window showing the pipeline stage output:

```
[Pipeline] stage
[Pipeline] { (Imprimir parametro APELLIDO)
[Pipeline] echo
Suárez
[Pipeline] echo
Suárez
[Pipeline] echo
Suárez
[Pipeline] }
[Pipeline] // stage
```

Se pueden poner más tipos de Parámetros:

RECUERDA: Se debe construir por primera vez para que se puedan empezar a usar los parámetros recien configurados.

```
pipeline {
    agent any
    parameters {
        string(name:'APELIDO', description:'Ingrese su Apellido')
        text(name:'BIOGRAFIA', description:'Ingrese su Biografía')
        booleanParam(name:'BOOLEANO', description:'Parametro Booleano')
        choice(name:'CIUDADES', choices:['Huelva','Sevilla','Córdoba','Cádiz'],
        description:'Selecciona una ciudad')
        password(name:'CONTRASENIA', description:'Ingrese su contraseña')
    }
    environment {
        CSA_INSTAGRAM="@csa_insta"
    }
}
```

The screenshot shows the Jenkins Pipeline configuration page for a pipeline named "Segundo Pipeline". The configuration code is displayed at the top, defining parameters forApellido, Bioografia, Booleano, Ciudad, and Contrasenia. Below the code, the pipeline stages are listed. The first stage, "BIOGRAFIA", has an input field labeled "Ingrese su Biografía". The second stage, "BOOLEANO", is a checkbox labeled "Parametro Booleano". The third stage, "CIUDADES", is a dropdown menu titled "Selecciona una ciudad" with options Huelva, Sevilla, Córdoba, and Cádiz, where "Huelva" is selected. At the bottom of the stage, there are "Ejecución" and "Cancel" buttons.

```

1  pipeline {
2      agent any
3      parameters {
4          string(name:'APELIDO', description:'Ingrese su Apellido')
5          text(name:'BIOGRAFIA', description:'Ingrese su Biografía')
6          booleanParam(name:'BOOLEANO', description:'Parametro Booleano')
7          choice(name:'CIUDADES', choices:['Huelva','Sevilla','Córdoba','Cádiz'], description:'Selecciona una ciudad')
8          password(name:'CONTRASENIA', description:'Ingrese su contraseña')
9      }
10     environment {
11         CSA_INSTAGRAM="@csa_insta"
12     }
13     stages {

```

Jenkins / Pipeline Projects / Segundo Pipeline

BIOGRAFIA
Ingrese su Biografía

BOOLEANO
Parametro Booleano

CIUDADES
Selecciona una ciudad

Huelva
Sevilla
Córdoba
Cádiz

Ejecución Cancel

11. Curso de Jenkins - Creación y uso de credenciales

Administrar Jenkins → Credentials → System → Global Credentials

The screenshot shows the Jenkins Global Credentials management interface. At the top, there's a navigation bar with links for Jenkins, Administrar Jenkins, Credentials, System, and Global credentials (unrestricted). Below the navigation is a header titled "Global credentials (unrestricted)". A sub-header below it says "Credentials that should be available irrespective of domain specification to requirements matching." There's a table with columns for ID, Name, Kind, and Description. A message at the bottom of the table area says "This credential domain is empty. How about adding some credentials?". At the bottom left, there are icons for S (Secret), M (Masked), and L (Large). On the right side, there's a blue button labeled "+ Add Credentials".

Agregar credenciales:

Tipos de credenciales:

New credentials

Kind

Username with password

Username with password

GitHub App

GitLab API token

SSH Username with private key

Secret file

Secret text

Certificate

Muy Importante el ID, no se puede modificar. Se usará en el Pipeline.

New credentials

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

prueba02

 Treat username as secret ?

Password ?

ID ?

id_de_prueba_02

An internal unique ID by which these credentials are identified from job

Description ?

Credenciales de Prueba02

Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
	id_de_prueba_02	prueba02/***** (Credenciales de Prueba02)	Username with password Credenciales de Prueba02

Icono: S M L

Secret text

New credentials

Kind

Secret text

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Secret

ID ?

secret_text_jajajaja

Description ?

Texto Secreto

Create

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description	
	id_de_prueba_02	prueba02***** (Credenciales de Prueba02)	Username with password	
	secret_text_jajajaja	Texto Secreto	Secret text	

Al hacer una nueva Pipeline con el script de la Segunda, nos dará un error por el tema de los parámetros. Solo hay que actualizar la página.

✓ Tercer Pipeline

Enlaces permanentes

- "Última ejecución (#2) hace 25 Seg"
- "Última ejecución estable (#2) hace 25 Seg"
- "Última ejecución correcta (#2) hace 25 Seg"
- "Última ejecución fallida (#1) hace 4 Min 16 Seg"
- "Última ejecución fallida (#1) hace 4 Min 16 Seg"
- "Last completed build (#2) hace 25 Seg"

¿Cómo meter credenciales pero sin que el técnico tenga que saber la contraseña?

<https://www.jenkins.io/doc/pipeline/steps/credentials-binding/> Como usar las diferentes credenciales dentro un Pipeline

o `usernamePassword`

Sets one variable to the username and one variable to the password given in the credentials.

Warning: if the Jenkins controller or agent node has multiple executors, any other build running concurrently on the same node will be able to read the text of the secret, for example on Linux using `ps e`.

▪ `usernameVariable : String`

Name of an environment variable to be set to the username during the build.

▪ `passwordVariable : String`

Name of an environment variable to be set to the password during the build.

▪ `credentialsId : String`

Credentials of an appropriate type to be set to the variable.

o `$class: 'VaultCertificateCredentialsBinding'`

```
pipeline {
    agent any
    parameters {
        string(name:'APELIDO', description:'Ingrese su Apellido')
        text(name:'BIOGRAFIA', description:'Ingrese su Biografía')
        booleanParam(name:'BOOLEANO', description:'Parametro Booleano')
        choice(name:'CIUDADES', choices:['Huelva','Sevilla','Córdoba','Cádiz'],
description:'Selecciona una ciudad')
        password(name:'CONTRASENIA', description:'Ingrese su contraseña')
    }
    environment {
        CSA_INSTAGRAM="@csa_insta"
    }
    stages {
        stage ("Saludo 1") {
            steps {
                echo "Hola desde Saludo 1"
            }
        }
        stage ("Saludo 2") {
            steps {
                echo "Hola desde Saludo 2, paso 1"
                echo "Hola desde Saludo 2, paso 2"
            }
        }
        stage ("Imprimir variable de entorno") {
            steps {
                echo "${env.CSA_INSTAGRAM}"
                // echo "${CSA_INSTAGRAM}"
                // echo "$CSA_INSTAGRAM"

                sh "echo 'hola mundo'"
                sh "echo '${env.CSA_INSTAGRAM}'"
                sh "echo '${CSA_INSTAGRAM}'"
                sh "echo '$CSA_INSTAGRAM'"

            }
        }
        stage ("Imprimir variable de entorno definida en System") {
            steps {
                sh "echo '${env.JAMON}'"
                sh "echo '${JAMON}'"
                sh "echo '$JAMON'"

            }
        }
    }
}
```

```
stage ("Cambiar valor a variable de entorno definida previamente en documento Pipeline") {
    steps {
        script {
            def temp_CSA_INSTAGRAM = "@cristobal_insta"
            echo "${temp_CSA_INSTAGRAM}"
            // env.CSA_INSTAGRAM = "@cristobal_insta_modificado"
        }
    }
}
stage ("Imprimir parametro APELLIDO") {
    steps {
        echo "${params.APELLIDO}"
        echo "${APELLIDO}"
        // echo "$APELIDO"
    }
}
stage ("Imprimir credenciales") {
    steps {
        withCredentials([usernamePassword(credentialsId:'id_de_prueba_02',
usernameVariable:'CUSTOM_USER', passwordVariable:'CUSTOM_PASSWORD')]) {
            script {
                echo "El nombre de usuario es: $CUSTOM_USER y la contraseña es: $CUSTOM_PASSWORD"
            }
        }
        withCredentials([string(credentialsId:'secret_text_jajajaja',
variable:'MY_CUSTOM_SECRET')]) {
            script {
                echo "El valor de mi texto secreto es $MY_CUSTOM_SECRET"
            }
        }
    }
}
```

✓ Imprimir credenciales ⏱ 6

✓ El nombre de usuario es: prueba02 y la contraseña es: \${CUSTOM_PASSWORD} ▾

0 Warning: A secret was passed to "echo" using Groovy String interpolation, which is insecure.
1 Affected argument(s) used the following variable(s): [CUSTOM_PASSWORD]
2 See <https://jenkins.io/redirect/groovy-string-interpolation> for details.

3 El nombre de usuario es: prueba02 y la contraseña es: ****

✓ El valor de mi texto secreto es \${MY_CUSTOM_SECRET} ▾

0 Warning: A secret was passed to "echo" using Groovy String interpolation, which is insecure.
1 Affected argument(s) used the following variable(s): [MY_CUSTOM_SECRET]
2 See <https://jenkins.io/redirect/groovy-string-interpolation> for details.

3 El valor de mi texto secreto es ****

Pasa las credenciales pero nosotros no tenemos porque saberlas.

12. Curso de Jenkins - Clonando un Repositorio

Podemos hacerlo con un solo comando para automatizarlo

```
git clone https://username:password@github.com/username/repository.git
```

Creamos unas credenciales de usuario password y ponemos ambos en secreto.

The screenshot shows the Jenkins interface for a Pipeline Project named 'Cuarto Pipeline'. The current build is '#3'. The 'Allocate node : Start' step has been completed, and the 'Workspace' step is active. The 'Workspace' page displays the contents of the repository, including a .git folder and several files: .htaccess, biblioteca.sql, index.php, probando_commit.txt, README.md, and testdb.php. Each file is listed with its last modified date, size, and two small icons. Below the file list is a blue link labeled '(Descargar archivos en formato zip)' which allows for a single download of all files.

File	Last Modified	Size	Actions
.htaccess	7 dic 2025 15:30:17	132 B	🔗 🔍
biblioteca.sql	7 dic 2025 15:30:17	10.53 KiB	🔗 🔍
index.php	7 dic 2025 15:30:17	1.08 KiB	🔗 🔍
probando_commit.txt	7 dic 2025 15:30:17	21 B	🔗 🔍
README.md	7 dic 2025 15:30:17	158 B	🔗 🔍
testdb.php	7 dic 2025 15:30:17	240 B	🔗 🔍

Podremos descargarnos el repositorio

```
pipeline {
    agent any

    stages {

        stage ("Clear Workspace") {
            steps {
                sh "rm -f /*"
            }
        }

        stage ("Git Clone") {
            steps {
                withCredentials([usernamePassword(credentialsId:'GITHUB_CREDENTIALS',
usernameVariable:'GITHUB_USER', passwordVariable:'GITHUB_PASSWORD')]) {

                    sh "git clone
https://$GITHUB_USER:$GITHUB_PASSWORD@github.com/cristobal90palma/Bibliot
eca-IAW-2ASIR-Cristobal.git"
                }
            }
        }
    }
}
```

13. Curso de Jenkins - Trigger Job

Ejecutar Jobs dentro de otro Jobs.

Script :

```

1  pipeline {
2      agent any
3      parameters {
4          string(name: 'NOMBRE_PIPELINE')
5          string(name: 'ID_JOB')
6      }
7      stages {
8          stage ("Hola mundo") {
9              steps {
10                 script {
11                     sh "echo 'hola mundo'"
12                 }
13             }
14         }
15         stage("Print params") {
16             steps {
17                 script {
18                     sh "echo $NOMBRE_PIPELINE"
19                     sh "echo $ID_JOB"
20                 }
21             }
22         }
23     }
24 }
```

Quinto Pipeline:

```

pipeline {
    agent any
    parameters {
        string(name: 'NOMBRE_PIPELINE')
        // FIX: Changed 'strin' to 'string'
        string(name: 'ID_JOB')
    }
    stages {
        // FIX: Changed 'stagre' to 'stage'
        stage ("Hola mundo"){
            steps {
                // The 'script' block is optional here, but safe to keep.
                script{
                    sh "echo 'hola mundo'"
                }
            }
        }
        // FIX: Changed 'stagre' to 'stage'
        stage("Print params"){
            steps {
```

```
// The 'script' block is optional here, but safe to keep.  
script {  
    // Parameters are accessed via the global 'params' object  
    // but since they have simple names, direct shell access often works too.  
    sh "echo The pipeline name is: ${params.NOMBRE_PIPELINE}"  
    sh "echo The job ID is: ${params.ID_JOB}"  
}  
}  
}  
}  
}
```

El trabajo como tal viene en otro JOB, para este caso usaremos el CUARTO.
El “build job” tiene que coincidir con el nombre del quinto pipeline:

```
pipeline {  
    agent any  
  
    stages {  
        stage ("Clear Workspace") {  
            steps {  
                sh "rm -rf /*"  
            }  
        }  
  
        stage ("Git Clone") {  
            steps {  
                withCredentials([usernamePassword(credentialsId:'GITHUB_CREDENTIALS',  
usernameVariable:'GITHUB_USER', passwordVariable:'GITHUB_PASSWORD')]) {  
                    sh "git clone  
https://$GITHUB_USER:$GITHUB_PASSWORD@github.com/cristobal90palma/Biblioteca-  
IAW-2ASIR-Cristobal.git"  
                }  
            }  
        }  
  
        // FIX 1: Added curly braces to properly define the stage block  
        stage("trigger job") {  
            steps {  
                build job: 'Quinto Pipeline',  
                parameters: [  
                    // FIX 2: Removed the extra colon (:) from the string parameter definition  
                    string(name: 'NOMBRE_PIPELINE', value: 'Cuarto Pipeline'),  
                    string(name: 'ID_JOB', value: '101011110110111011011')  
                ]  
            }  
        }  
    }  
}
```

Como hacemos para pasarle valores dinámicos.

<http://localhost:8080/env-vars.html/>

```
stage("trigger job") {
    steps {
        build job: 'Quinto Pipeline',
        parameters: [
            // FIX 2: Removed the extra colon (:) from the string parameter definition
            string(name: 'NOMBRE_PIPELINE', value: "$JOB_NAME"),
            string(name: 'ID_JOB', value: "$BUILD_ID")
        ]
}
```

Guardamos y corremos de nuevo.

Esto se ve en el Quinto Pipeline

```
+ echo The pipeline name is:
The pipeline name is:
[Pipeline] sh
+ echo The job ID is: 9
The job ID is: 9
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

14. Curso de Jenkins - JenkinsFile y Repositorios Git

No te lies y mete el fichero jenkinfile en github desde el navegador web.

Copia el contenido del Primer Pipeline en un fichero llamado Jenkinfile en un repositorio de GitHub.

Ahora nos vamos a la configuración del Nuevo Pipeline (Sexto Pipeline)

The screenshot shows the Jenkins Pipeline configuration interface. The top navigation bar indicates 'Projects' > 'Sexto Pipeline' > 'Configuration'. The main section is titled 'Pipeline' with the sub-section 'Definition' selected. It says 'Define your Pipeline using Groovy directly or pull it from source control.' Below this, the 'SCM' dropdown is set to 'Git'. Under 'Repositories', there is one entry with 'Repository URL' set to 'https://github.com/cristobal90palma/gitiaw16102025' and 'Credentials' set to 'Credenciales de GitHub de mentira'. A blue box highlights the 'Credentials' dropdown.

Especificamos la rama:

The screenshot shows the 'Branch Specifier (blank for 'any')' field, which contains the value '/main'. A blue box highlights this field.

Le indicamos el nombre del Jenkinfile, que puede ser cualquiera:

En mi caso es Jenkinfile, sin "s".

Script Path ?

Lightweight checkout ?

Si el archivo estuviese en una carpeta hacemos así:

Script Path ?

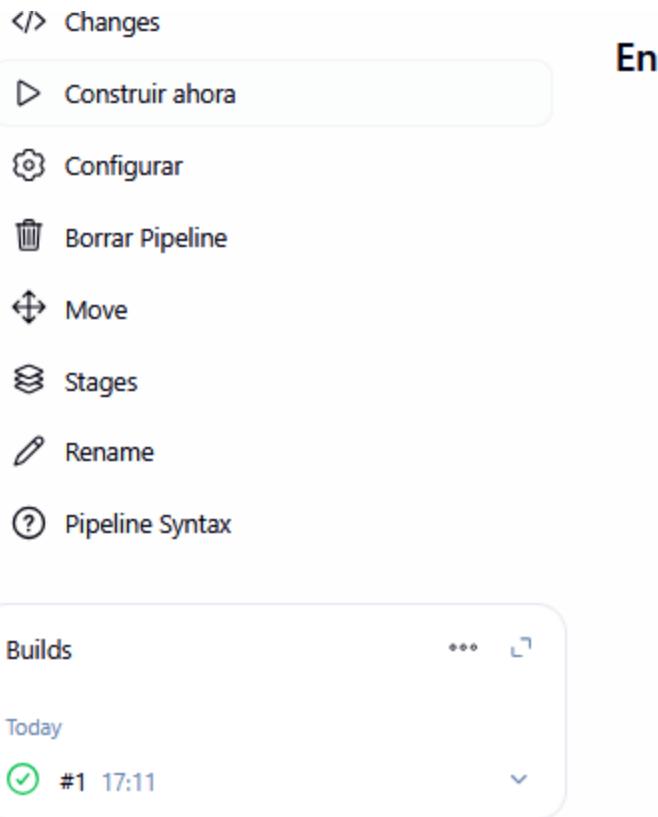
Lightweight checkout ?

[Pipeline Syntax](#)

Aplicamos y guardamos.

Cuando le demos a construir, Jenkins va al repositorio, lo descarga y lo usa.

Funciona



```
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential GITHUB_CREDENTIALS
Cloning the remote Git repository
Cloning repository https://github.com/cristobal90palma/gitiaw16102025
> git init /var/jenkins_home/workspace/Sexto Pipeline # timeout=10
Fetching upstream changes from https://github.com/cristobal90palma/gitiaw16102025
> git --version # timeout=10
> git --version # 'git version 2.47.3'
using GIT_ASKPASS to set credentials Credenciales de GitHub de mentira
> git fetch --tags --force --progress -- https://github.com/cristobal90palma/gitiaw16102025 +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/cristobal90palma/gitiaw16102025 # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 398a097d0774c817f9567f00cc19cd1bb7162ad1 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 398a097d0774c817f9567f00cc19cd1bb7162ad1 # timeout=10
Commit message: "Create Jenkins pipeline with echo stages"
First time build. Skipping changelog.
[Pipeline]
[Pipeline] // stage
[Pipeline] withEnv
```

En la parte de abajo se ve como ejecuta las instrucciones que hay en el Jenkinsfile