

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
e4beac64dffa: Pull complete
744b4792e083: Pull complete
de160a647174: Pull complete
dc2a77f462ea: Pull complete
c27509c3e53b: Pull complete
05a7d9a8b608: Pull complete
Digest: sha256:ddcec7ebee31abeb9aa864510a19874d03d53948b040e19461e3432b
e5455b1f
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	Q Credentials Binding	** commons-lang3 v3.x Jenkins API ** Ionicons API
Q Timestampers	Q Workspace Cleanup	Q Ant	Q Gradle	Folders OWASP Markup Formatter ** ASM API ** JSON Path API ** Structs ** Pipeline: Step API ** commons-text API ** Token Macro
Q Pipeline	Q GitHub Branch Source	Q Pipeline: GitHub Groovy Libraries	Q Pipeline Graph View	Build Timeout ** bouncycastle API ** Credentials ** Plain Credentials
Q Git	Q SSH Build Agents	Q Matrix Authorization Strategy	Q LDAP	
Q Email Extension	Q Mailer	Q Dark Theme		

Jenkins 2.528.2

** - required dependency

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.

← → ↺

🔒 📄 http://localhost:8080

 **Jenkins**

+ Nueva Tarea

📁 Historial de trabajos

Trabajos en la cola

No hay trabajos en la cola

Estado del ejecutor de construcciones

0/2

¡Bienvenido a Jenkins!

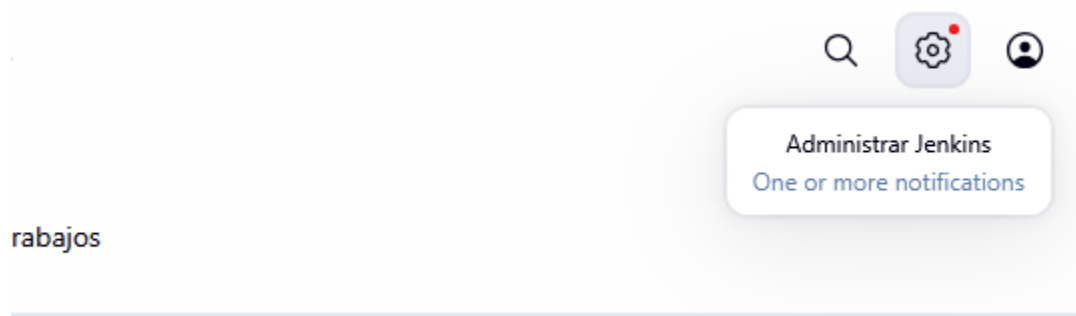
This page is where your Jenkins jobs will be displayed or start building a software project.

Start building your software project

Create a job

04. Curso de Jenkins - Administrar Jenkins





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



System Configuration





-  **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.



+ Create User

ie 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

User ID ↓	
	admin
	test

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

Usuarios autenticados tienen privilegios para todo

Configuración de seguridad

Cualquiera puede hacer cualquier acción

Estrategia de seguridad para el proyecto

Modo 'legacy'

Usuarios autenticados tienen privilegios para todo

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

Autorización

Configuración de seguridad

Global

Usuario/Grupo	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"/>

Add user

Add group

Guardar

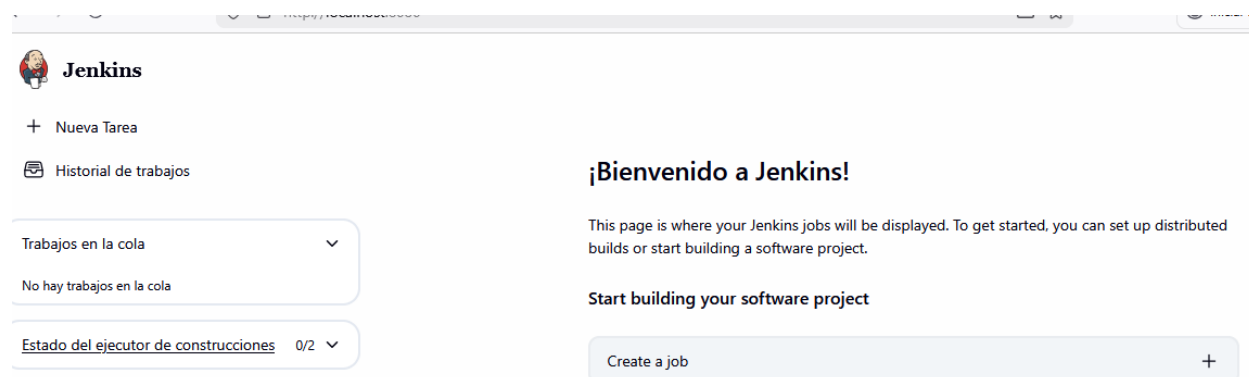
Apply

06. Curso de Jenkins - Job Free Style

Los Jobs es la columna vertebral de Jenkins.

Panel de Control → Nueva tarea

O Create a Job



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:

Todo	+					
S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración	
...	☀	Prueba01	N/D	N/D	N/D	▶

Y también podremos configurar el JOB



Jenkins / Prueba01 ▾ / Configuration

Configure



General



Configurar el origen del código fuente



Triggers



Environment



Build Steps



Acciones para ejecutar después.

General

Descripción

Plain text [Visualizar](#)

☐

Desechar ejecuciones antiguas ?

☐

Esta ejecución debe parametrizarse ?

☐

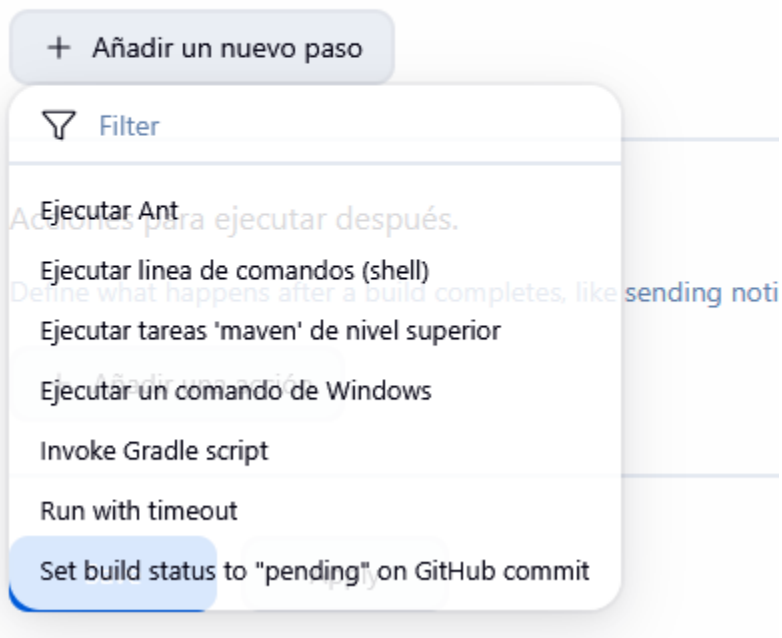
GitHub project

GitHub Connection

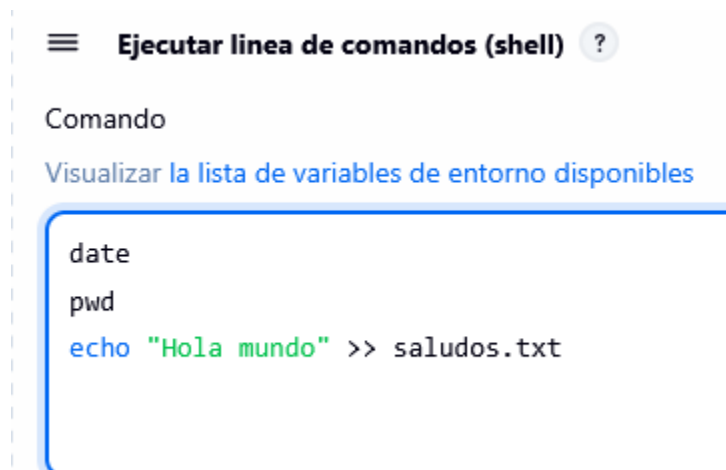
Nos vamos a “Build Steps” y le damos a “Ejecutar linea de comandos (Shell)”.

Build Steps

Automate your build process with ordered tasks like code com



Escribimos los comandos



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 comandos 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.

The screenshot shows the Jenkins web interface. At the top, it says 'Jenkins / Prueba01 / Workspace of Prueba01 on principal'. Below this, there are several buttons: 'Estado Actual', 'Cambios', 'Zona de Trabajo' (highlighted), 'Limpiar el espacio de trabajo', and 'Construir ahora'. To the right, under the heading 'Workspace of Prueba01 on principal', there is a search bar and a list of files: 'hola_caracola.txt' (7 dic 2025 9:57:27, 0 B) and 'saludos.txt' (7 dic 2025 9:57:27, 22 B). Below the file list is a link '(Descargar archivos en formato zip)'. At the bottom, there is a terminal window showing the output of a command: 'Hola mundo' and '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 "hackedo" >> has_sido_hackedo.txt
jenkins@9eab68d6b058:~/workspace/Prueba01$ cat has_sido_hackedo.txt
hackedo
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_hackedo.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_hackedo.txt
hackedo
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.



Jenkins

+ Nueva Tarea

Historial de trabajos

Nueva vista

Todo

+

Nueva vista

Nombre de vista

Free Style Projects

Type



Lista de vistas

Mostrar proyectos como una lista simple. Puedes elegir qué proyectos mostrar en cada vista.



Mi vista

Esta vista muestra automáticamente todas las tareas a las que el usuario puede acceder.

Create

Editar la vista

Nombre

Free Style Projects

Descripción

Describe the purpose of this view.

Estos son los proyectos Free Style Jobs

View And ViewEdit

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.


Jenkins / Free Style Projects

[+ Nueva Tarea](#)

[Historial de trabajos](#)



[Editar vista](#)

[Borrar vista](#)

Trabajos en la cola





Estos son los proyectos Free Style Jobs

Free Style Projects Todo +

S	W	Nombre ↓	Últi
		Prueba01	21 M

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


Free Style Projects Todo +

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

Free Style Projects Pipeline Projects Todo +



Esta vista no tiene ningún proyecto asociado.

[Añadir.](#)

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

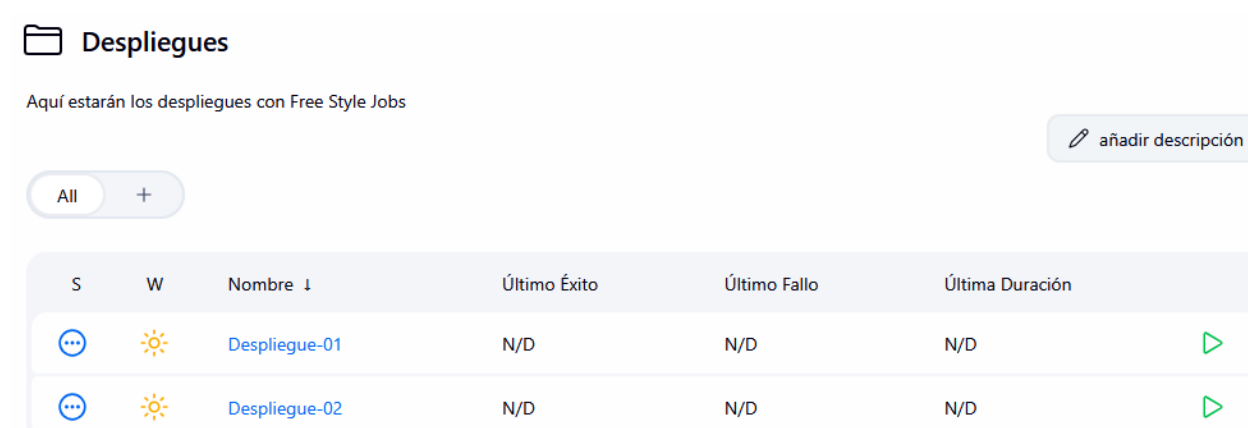
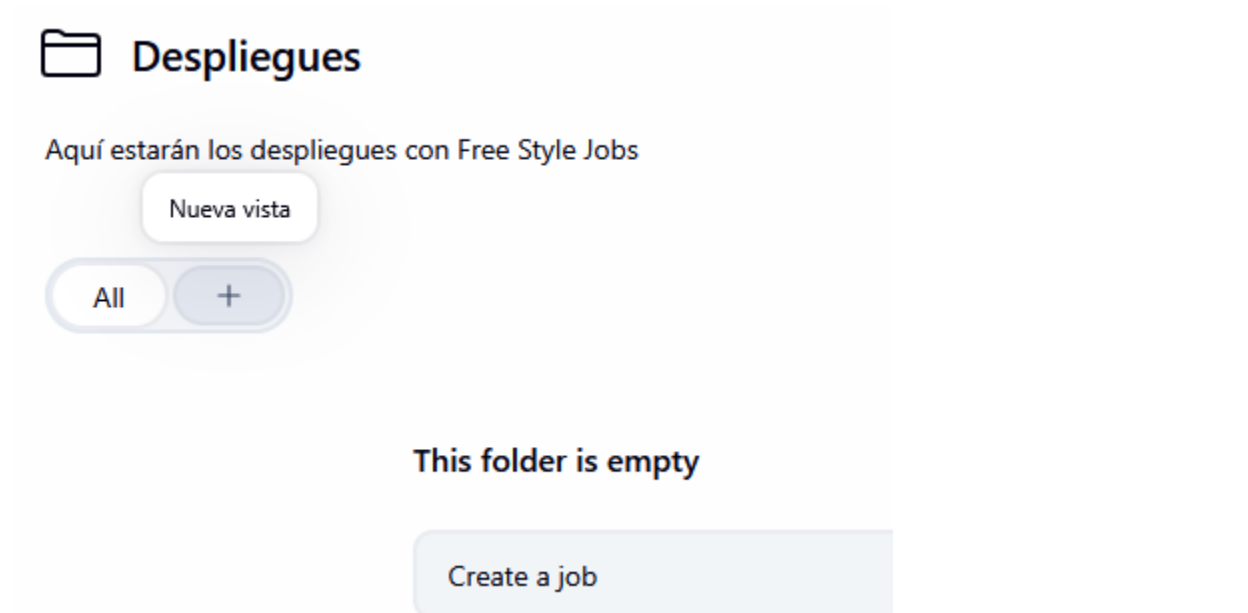
Pipeline Projects

Todo

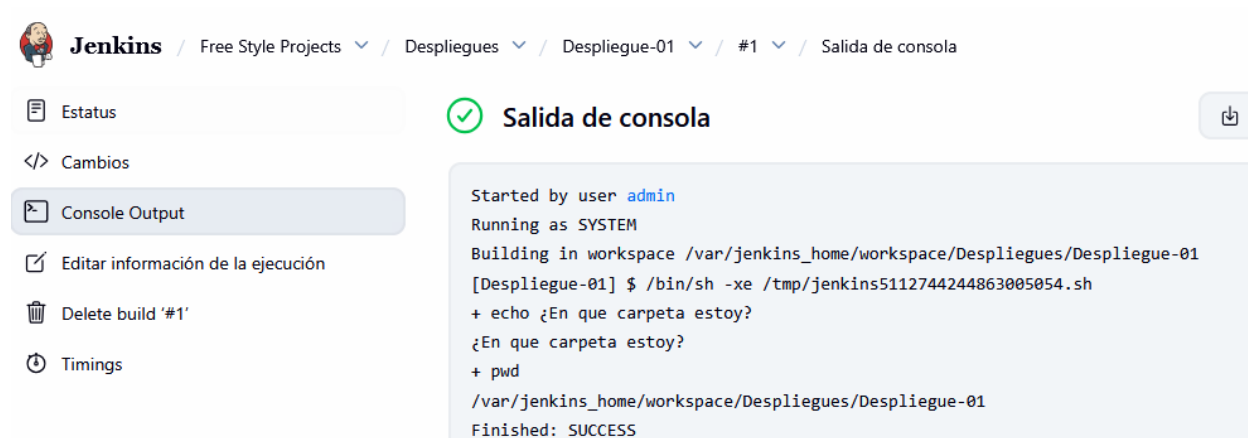


S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración
		Despliegues	N/D	N/D	N/D
		Prueba01	37 Min #4	34 Min #5	22 Ms
		Prueba02	9 Min 5 Seg #1	N/D	91 Ms

Dentro de la carpeta Podemos seguir creando vistas y jobs.



El concepto es el mismo.



08. Curso de Jenkins - Variables de entorno

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

Triggers
 Environment
 Build Steps
 Acciones para ejecutar después.

Ejecutar línea de comandos (shell) ?

Comando

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

Jenkins /

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 pi 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" 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 number.

CHANGE_URL

For a multibranch project corresponding to some kind of change request, this will be set to the change URL, if supported; else, it will be empty.

CHANGE_TITLE

For a multibranch project corresponding to some kind of change request, this will be set to the change title, if supported; else, it will be empty.

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

Ejecutar línea 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 línea 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.



Jenkins / Pipeline Projects ▾ / Nuevo Tarea

Nuevo Tarea

Enter an item name

Select an item type



Crear un proyecto de estilo libre

Classic, general-purpose job type that checks out from up to one source code repository and executes a series of steps 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 orquestación de tareas de tipo freestyle.



Crear un proyecto multi-configuración


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



Add to current view



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


Jenkins / Pipeline Projects / Primer Pipeline / Configuration

Configure

- General
- Triggers
- Pipeline**
- Advanced

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

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 }

```

Save

Apply

```

pipeline {
  agent any

  stages {
    stage ("Saludo 1") {
      steps {
        echo "Hola desde Saludo 1"
      }
    }
  }
}

```




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



 Search

✓ Saludo 1 0,66s

✓ Saludo 1

✓ Hola desde Saludo 1 ▾

0 Hola desde Saludo 1

cript :

```
1 pipeline {
2     agent any
3
4     stages {
5         stage ("Saludo 1") {
6             steps {
7                 echo "Hola desde Saludo 1"
8             }
9         }
10        stage ("Saludo 2") {
11            steps {
12                echo "Hola desde Saludo 2, paso 1"
13                echo "Hola desde Saludo 2, paso 2"
14            }
15        }
16    }
17 }
```

✓ < #3

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

Graph



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 ▾

0 @csa_insta

✓ @csa_insta ▾

0 @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 "echo '${env.CSA_INSTAGRAM}'"
    sh "echo '${CSA_INSTAGRAM}'"
    sh "echo '$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 - Parametros 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 '$JAMON'"
    }
}
stage ("Cambiar valor a variable de entorno definida previamente en documento
Pipeline") {
    steps {
        script {
            CSA_INSTAGRAM="@crisobal_insta"
            echo "${CSA_INSTAGRAM}"
        }
    }
}
}
}

```

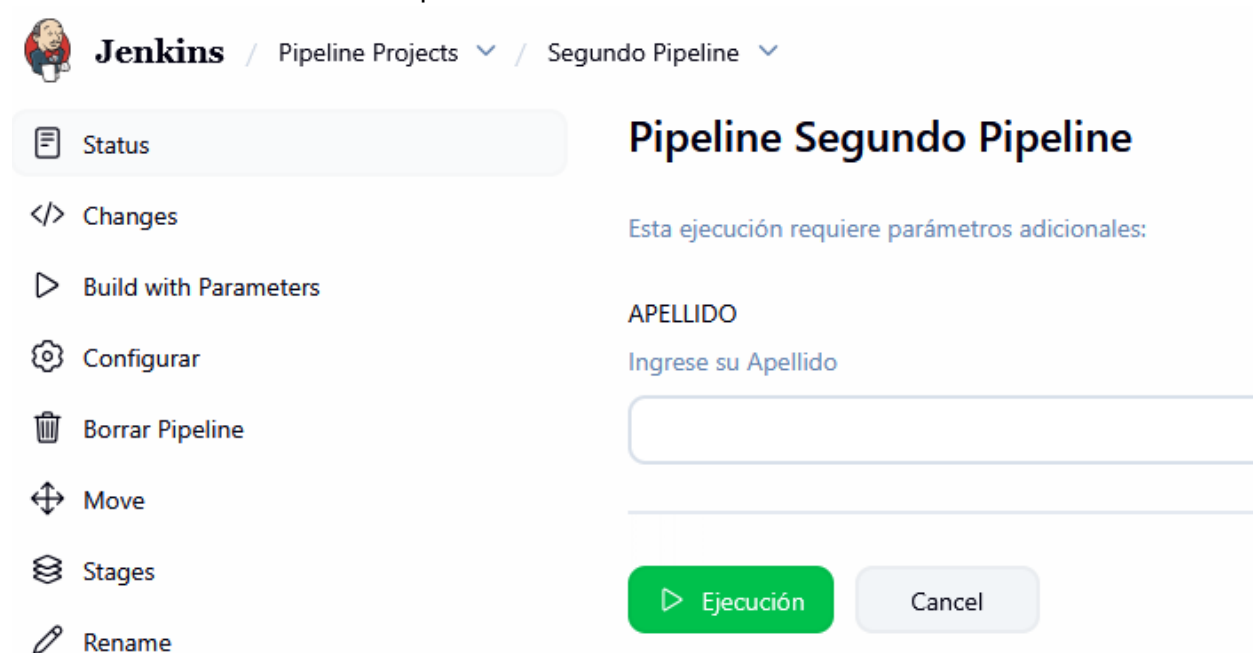
¿Cómo hacer para agregar un valor manualmente antes de que se ejecute el Pipeline?
Encima de Environment escribes parameters.

```

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

```

Ahora nos saldrá: “Build with parameters”.



The screenshot shows the Jenkins web interface. At the top, the breadcrumb navigation is 'Jenkins / Pipeline Projects / Segundo Pipeline'. On the left sidebar, there are several menu items: 'Status' (selected), 'Changes', 'Build with Parameters', 'Configurar', 'Borrar Pipeline', 'Move', 'Stages', and 'Rename'. The main content area is titled 'Pipeline Segundo Pipeline'. Below the title, it says 'Esta ejecución requiere parámetros adicionales:'. There is a parameter named 'APELLIDO' with the label 'Ingrese su Apellido' and an empty text input field. At the bottom right, there are two buttons: 'Ejecución' (green) and 'Cancel' (grey).

A partir de entonces podremos usar la información que queramos meter en el parámetro en las ordenes del Pipeline.

```
stage ("Imprimir parametro APELLIDO") {
    steps {
        echo "${params.APELLIDO}"
        echo "${APELLIDO}"
        echo "$APELLIDO"
    }
}
```

```
[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 recién configurados.

```
pipeline {
  agent any
  parameters {
    string(name:'APELLIDO', 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 interface. The left sidebar displays the pipeline configuration tree with the following structure:

- 1 pipeline {
 - 2 agent any
 - 3 parameters {
 - 4 string(name:'APELLIDO', 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 {

The right sidebar shows the execution interface for the 'Segundo Pipeline' project. It displays the following parameters:

- BIOGRAFIA**: Ingrese su Biografía (Text input field)
- BOOLEANO**: Parametro Booleano (Checkbox)
- CIUDADES**: Selecciona una ciudad (Dropdown menu with options: Huelva, Sevilla, Córdoba, Cádiz)

At the bottom, there are buttons for 'Ejecución' (Execution) and 'Cancel'.

11. Curso de Jenkins - Creación y uso de credenciales

Administrar Jenkins → Credentials → System → Global Credentials

Global credentials (unrestricted)

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

ID	Name	Kind	Description
This credential domain is empty. How about adding some credentials?			

Icono: S M L

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	Credenciales de Prueba02	
 secret_text_jajajaja	Texto Secreto	Secret text	Texto Secreto	

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

◦ `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.

◦ `$class: 'VaultCertificateCredentialsBinding'`

```

pipeline {
  agent any
  parameters {
    string(name:'APELLIDO', 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 "${APELLIDO}"
        // echo "$APELLIDO"
    }
}
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 ****
```

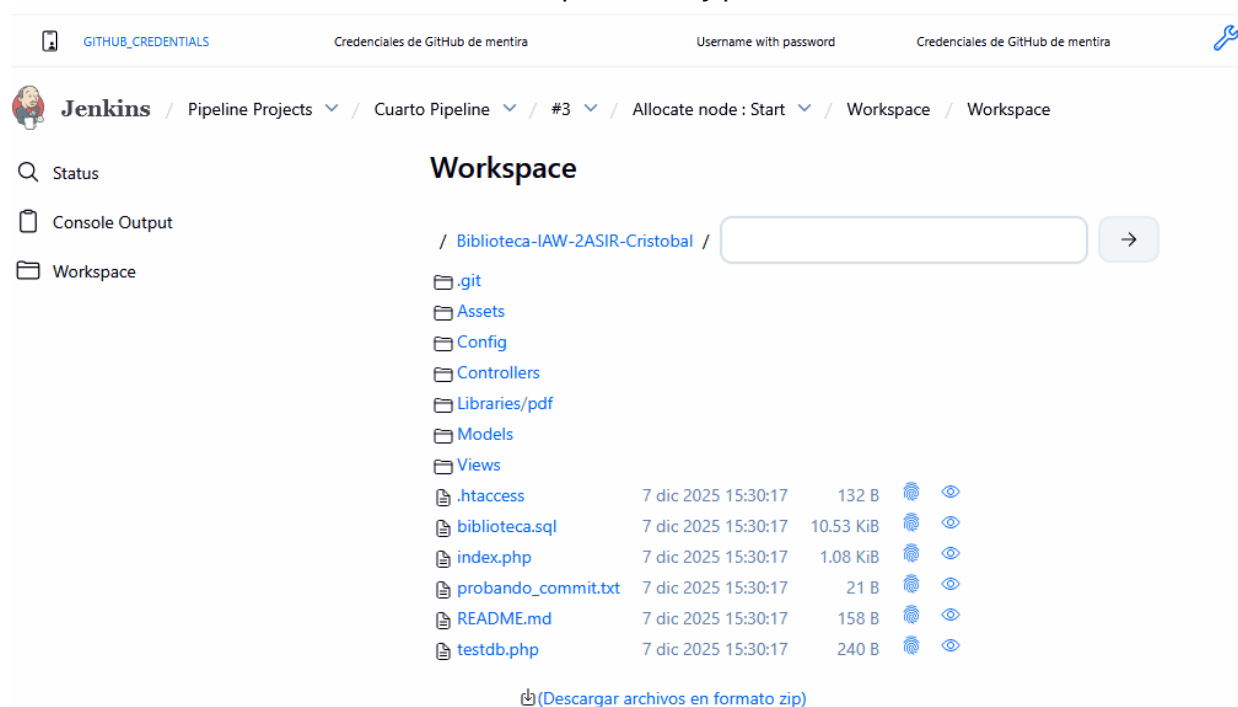
Pasa las credenciales pero nosotros no tenemos porque saberlas.

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 web interface. At the top, there's a navigation bar with a 'GITHUB_CREDENTIALS' link and several tabs: 'Credenciales de GitHub de mentira', 'Username with password', and 'Credenciales de GitHub de mentira'. Below this is a breadcrumb trail: 'Jenkins / Pipeline Projects / Cuarto Pipeline / #3 / Allocate node : Start / Workspace / Workspace'. On the left sidebar, there are links for 'Status', 'Console Output', and 'Workspace'. The main area is titled 'Workspace' and shows a file tree for the path '/ Biblioteca-IAW-2ASIR-Cristobal /'. The tree includes folders like '.git', 'Assets', 'Config', 'Controllers', 'Libraries/pdf', 'Models', and 'Views'. Below these are several files with their details: '.htaccess' (132 B), 'biblioteca.sql' (10.53 KiB), 'index.php' (1.08 KiB), 'probando_commit.txt' (21 B), 'README.md' (158 B), and 'testdb.php' (240 B). Each file entry shows its last modified date and time (7 dic 2025 15:30:17) and has icons for download and view. At the bottom, there's a link to '(Descargar archivos en formato zip)'.

Podremos descargarlos 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.

```

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.NOMPRE_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

```
[Pipeline] sh
+ 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 jenkinsfile en github desde el navegador web.

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

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

The screenshot shows the Jenkins Pipeline Configuration page for a project named 'Sexto Pipeline'. The breadcrumb navigation at the top reads 'Projects / Sexto Pipeline / Configuration'. The page title is 'Pipeline', with a subtitle 'Define your Pipeline using Groovy directly or pull it from source control.' Under the 'Definition' section, the 'Pipeline script from SCM' option is selected. In the 'SCM' section, 'Git' is chosen. The 'Repositories' section contains one entry with the 'Repository URL' set to 'https://github.com/cristobal90palma/gitiaw16102025'. The 'Credentials' dropdown is set to 'Credenciales de GitHub de mentira', with an '+ Add' button next to it. An 'Avanzado' (Advanced) dropdown is visible at the bottom of the configuration area.

Especificamos la rama:

This screenshot shows the 'Branch Specifier (blank for 'any')' field. The text '*/main' is entered into the input box.

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

En mi caso es Jenkinsfile, sin “s”.

Script Path ?

Jenkinsfile

☒ Lightweight checkout ?

Si el archivo estuviese en una carpeta hacemos así:

Script Path ?

/ClaseSeis/Jenkinsfile

☒ Lightweight checkout ?

[Pipeline Syntax](#)

Aplicamos y guardamos.

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

Funciona

</> Changes

▶ Construir ahora

⚙ Configurar

🗑 Borrar Pipeline

↕ Move

📁 Stages

✎ Rename

❓ Pipeline Syntax

En

Builds

Today

✅ #1 17:11

```
[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