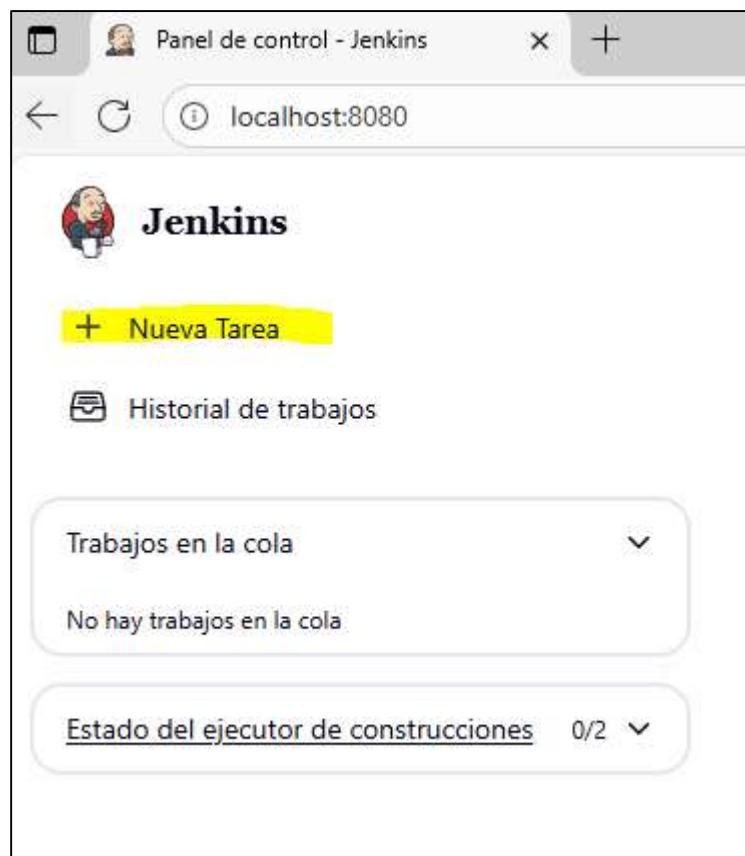


TAREAS EN JENKIS

1.	TAREA DE COMANDOS DE WINDOWS.....	1
2.	TAREA DE MAVEN	9
3.	PIPELINE.....	18

1. TAREA DE COMANDOS DE WINDOWS



localhost:8080/view/all/new/job

Jenkins / Todo / 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 SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Pipeline**  Gestiona actividades de larga duración que pueden abarcar varios agentes de construcción. Apropiado para construir pipelines (conocidas anteriormente como workflows) y/o para la organización de actividades complejas que no se pueden articular fácilmente con tareas de tipo freestyle.
- Crear un proyecto multi-getConfiguración**  Adequado para proyectos que requieran un gran número de configuraciones diferentes, como testear en múltiples entornos, ejecutar sobre plataformas concretas, etc.
- Folder**  Crea un container que almacena ítems anidados. Útil para agrupar cosas juntas. Diferente a view, que es solo un filtro, un folder crea un espacio de nombre separado, así puedes tener múltiples cosas con el mismo nombre, siempre y cuando estén en diferentes carpetas.
- Multibranch Pipeline**  Crea un conjunto de Pipeline projects según detectados en uno SCM repository.
- Organization Folder**  Crea un conjunto de subcarpetas para proyectos multibranch por escaneo de repositorios.

OK

Jenkins / Tarea comandos basica / Configuration

Configure

General

Descripción

Esta tarea muestra por consola la fecha actual, una descripción y crea la carpeta trabajo inicial.

Plain text [Visualizar](#)

Desechar ejecuciones antiguas [?](#)

Esta ejecución debe parametrizarse [?](#)

GitHub project

Throttle builds [?](#)

Lanzar ejecuciones concurrentes en caso de ser necesario [?](#)

Avanzado [▼](#)

Configurar el origen del código fuente

Connect and manage your code repository to automatically pull the latest code for your builds.

Ninguno

Git [?](#)

localhost:8080/job/Tarea%20comandos%20basica/configure

Jenkins / Tarea comandos basica / Configuration

Configure

General

Triggers

Environment

Build Steps

Acciones para ejecutar después.

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- Lanzar ejecuciones remotas (ejem: desde 'scripts')
- Construir tras otros proyectos
- Consultar repositorio (SCM)
- Ejecutar periódicamente
- GitHub hook trigger for GITScm polling

Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

- Delete workspace before build starts
- Use secret text(s) or file(s)
- Abortar la ejecución si se atasca
- Add timestamps to the Console Output
- Inspect build log for published build scans
- With Ant

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

+ Añadir un nuevo paso

Acciones para ejecutar después.

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

+ Añadir una acción

Save **Apply**

localhost:8080/job/Tarea%20comandos%20basica/configure

Jenkins / Tarea comandos basica / Configuration

Configure

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Ejecutar línea de comandos (shell)

Comando

Visualizar la lista de variables de entorno disponibles

```
DATE /T  
echo "Mi primer trabajo en Jenkins que muestra por consola la fecha actual, una descripción y crea la carpeta trabajo_inicial."  
mkdir trabajo_inicial
```

Avanzado ▾

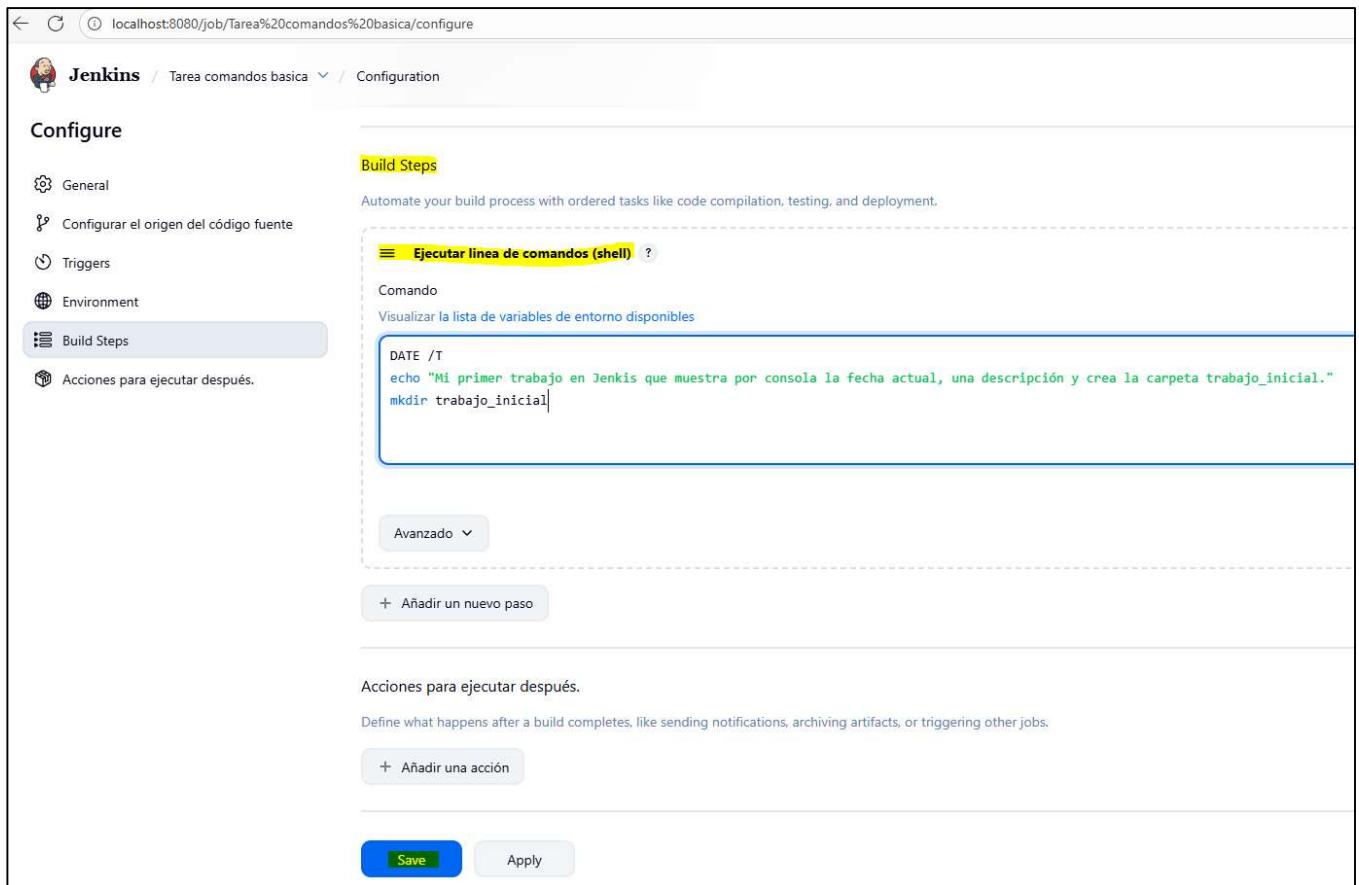
+ Añadir un nuevo paso

Acciones para ejecutar después.

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

+ Añadir una acción

Save **Apply**



localhost:8080

Jenkins

+ Nueva Tarea

Historial de trabajos

Todo +

Trabajos en la cola

No hay trabajos en la cola

Estado del ejecutor de construcciones 0/2

Último Éxito Último Fallo Última Duración

S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración
		Tarea comandos basica	N/D	N/D	N/D

Icono: S M L



 Jenkins / Tarea comandos basica

Estado Actual  **Tarea comandos basica**

</> Cambios
Zona de Trabajo
Construir ahora
Configurar
Borrar Proyecto
Rename

Builds ... 

Filter Today #1 16:51

Esta tarea muestra por consola la fecha actual, una descripción y crea la carpeta trabajo_inicial.

Enlaces permanentes

- ["Última ejecución \(#1\) hace 4,1 Seg"](#) ↗
- ["Última ejecución fallida \(#1\) hace 4,1 Seg"](#)
- ["Última ejecución fallida \(#1\) hace 4,1 Seg"](#)
- ["Last completed build \(#1\) hace 4,1 Seg"](#)

 Jenkins / Tarea comandos basica #1

Estatus  **#1 (28 nov 2025, 16:51:54)**

</> Cambios
 Console Output  Iniciado por el usuario Patricia
 Editar información de la ejecución  This run spent:

- 32 Ms waiting;
- 0.28 Seg build duration;
- 0.31 Seg total from scheduled to completion.

 Delete build '#1'
 Timings

</> Sin cambios

Jenkins / Tarea comandos basica #1 / Salida de consola

Estatus Cambios Console Output Editar información de la ejecución Delete build '#1' Timings

Salida de consola

```
Lanzada por el usuario Patricia
Running as SYSTEM
Ejecutando en el espacio de trabajo C:\ProgramData\Jenkins\workspace\Tarea comandos basica
[Tarea comandos basica] $ sh -xe C:\Windows\TEMP\jenkins1139453778338072250.sh
El sistema no puede encontrar el fichero especificado
FATAL: la ejecución del comando ha fallado
java.io.IOException: CreateProcess error=2, El sistema no puede encontrar el archivo especificado
        at java.base/java.lang.ProcessImpl.create(Native Method)
        at java.base/java.lang.ProcessImpl.<init>(ProcessImpl.java:506)
        at java.base/java.lang.ProcessImpl.start(ProcessImpl.java:159)
        at java.base/java.lang.ProcessBuilder.start(ProcessBuilder.java:1126)
Caused: java.io.IOException: Cannot run program "sh" (in directory "C:\ProgramData\Jenkins\workspace\Tarea comandos basica"): CreateProcess error=2, El sistema no puede encontrar el archivo especificado
        at java.base/java.lang.ProcessBuilder.start(ProcessBuilder.java:1170)
        at java.base/java.lang.ProcessBuilder.start(ProcessBuilder.java:1089)
        at hudson.Proc$LocalProc.<init>(Proc.java:252)
        at hudson.Proc$LocalProc.<init>(Proc.java:221)
        at hudson.Launcher$LocalLauncher.launch(Launcher.java:995)
        at hudson.Launcher$ProcStarter.start(Launcher.java:507)
        at hudson.tasks.CommandInterpreter.perform(CommandInterpreter.java:144)
        at hudson.tasks.CommandInterpreter.perform(CommandInterpreter.java:92)
        at hudson.tasks.BuildStepMonitor$1.perform(BuildStepMonitor.java:20)
        at hudson.model.AbstractBuild$AbstractBuildExecution.perform(AbstractBuild.java:818)
        at hudson.model.AbstractBuild$AbstractBuildExecution.build(AbstractBuild.java:199)
        at hudson.model.AbstractBuild$AbstractBuildExecution.doRun(AbstractBuild.java:164)
        at hudson.model.AbstractBuild$AbstractBuildExecution.run(AbstractBuild.java:527)
        at hudson.model.Run.execute(Run.java:1866)
        at hudson.model.FreeStyleBuild.run(FreeStyleBuild.java:44)
        at hudson.model.ResourceController.execute(ResourceController.java:181)
        at hudson.model.Executor.run(Executor.java:468)
Build step 'Ejecutar linea de comandos (shell)' marked build as failure
Finished: FAILURE
```

Jenkins

+ Nueva Tarea Historial de trabajos

Trabajos en la cola

S	W	Nombre	Último Éxito	Último Fallo	Última Duración
		Tarea comandos basica	N/D	3 Min 37 Seg #1	0.28 Seg

Estado del ejecutor de construcciones 0/2

Icono: S M L

Jenkins / Administrar Jenkins

Administrar Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

System Configuration


System

Configurar variables globales y rutas.


Tools

Configure tools, their locations and installers.

**Jenkins**

/ Administrar Jenkins

/ System

 Preserve second fetch during checkout ? Add git tag action to jobs ?**Línea de comandos**

Ejecutable para la línea de comandos (shell) ?

C:\Windows\system32\cmd.exe

Extended E-mail Notification

SMTP server

SMTP Port

25

Avanzado ▾

Default user e-mail suffix ?

Avanzado ▾

/ Edited

Default Content Type ?

Save

Apply

← ⏪ (i) localhost:8080/job/Tarea%20comandos%20basica/

 Jenkins / Tarea comandos basica

Estado Actual Tarea comandos basica

Cambios Zona de Trabajo Construir ahora Configurar Borrar Proyecto Rename

Esta tarea muestra por consola la fecha actual, una descripción y crea la carpeta trabajo_inicial.

Enlaces permanentes

- Última ejecución (#2) hace 7,2 Seg
- Última ejecución estable (#2) hace 7,2 Seg
- Última ejecución correcta (#2) hace 7,2 Seg
- Última ejecución fallida (#1) hace 7 Min 25 Seg
- Última ejecución fallida (#1) hace 7 Min 25 Seg
- Last completed build (#2) hace 7,2 Seg

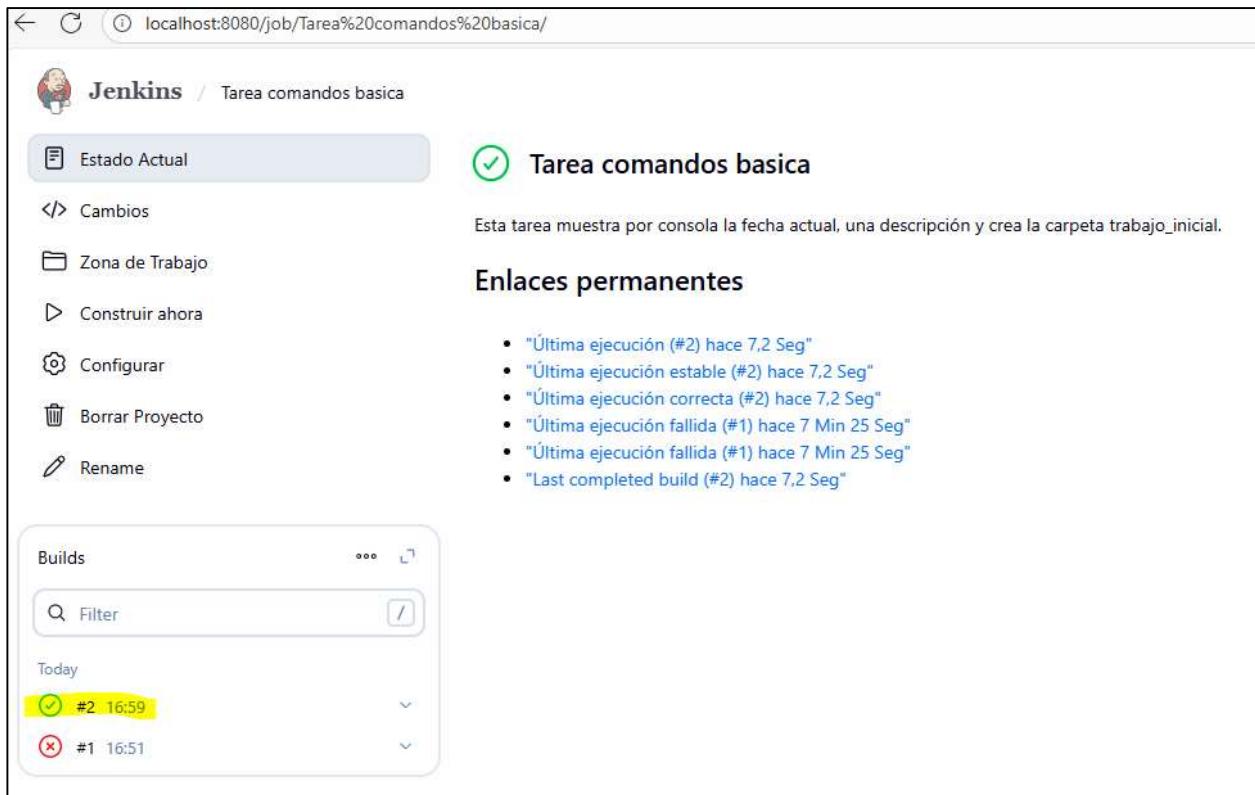
Builds

Filter

Today

#2 16:59

#1 16:51



2. TAREA DE MAVEN



The screenshot shows the "Nuevo Tarea" (New Job) creation dialog. At the top, there's a breadcrumb navigation: Jenkins / Todo / Nuevo Tarea. The main title is "Nuevo Tarea". A text input field asks "Enter an item name" with the value "Tarea maven" highlighted. Below this, a section titled "Select an item type" lists several options:

- Crear un proyecto de estilo libre**: Description: Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Pipeline**: Description: Gestiona actividades de larga duración que pueden abarcar varios agentes de construcción. Apropósito para construir pipelines (conocidas anteriormente como workflows) y/o para la organización de actividades complejas que no se pueden articular fácilmente con tareas de tipo freestyle.
- Crear un proyecto multi-configuración**: Description: Adequado para proyectos que requieren un gran número de configuraciones diferentes, como testear en múltiples entornos, ejecutar sobre plataformas concretas, etc.
- Folder**: Description: 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.
- Multibranch Pipeline**: Description: Creates a set of Pipeline projects according to detected branches in one SCM repository.
- Organization Folder**: Description: Creates a set of multibranch project subfolders by scanning for repositories.

At the bottom, there's a note: "If you want to create a new item from other existing, you can use this option: Copy from" followed by an "OK" button.



Jenkins

/ Tarea maven

/ Configuration

Configure

General

General

Configurar el origen del código fuente

Triggers

Environment

Build Steps

Acciones para ejecutar después.

Descripción

Ejecuta el comando de maven clean sobre un proyecto en el repositorio github

Plain text [Visualizar](#)

Desechar ejecuciones antiguas [?](#)

Esta ejecución debe parametrizarse [?](#)

GitHub project

Throttle builds [?](#)

Lanzar ejecuciones concurrentes en caso de ser necesario [?](#)

[Avanzado](#)

Configurar el origen del código fuente

Connect and manage your code repository to automatically pull the latest code for your builds.

Ninguno

Git [?](#)

Jenkins / Tarea maven / Configuration

Configure

Configurar el origen del código fuente

Connect and manage your code repository to automatically pull the latest code for your builds.

Ninguno

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

https://github.com/patriciasfo/biblioteca.git

Credentials [?](#)

patriciasfo/*********

Avanzado [▼](#)

+ Add Repository

Branches to build [?](#)

Branch Specifier (blank for 'any') [?](#)

*/master

+ Add Branch

Navegador del repositorio [?](#)

(Auto)

This screenshot shows the Jenkins configuration interface for a Maven task. The left sidebar lists options like General, Triggers, Environment, Build Steps, and Actions after build. The 'General' section is selected. The main area is titled 'Configurar el origen del código fuente' (Configure code source origin) with the sub-instruction 'Connect and manage your code repository to automatically pull the latest code for your builds.' A radio button for 'Ninguno' is unselected, while 'Git' is selected. Below this, there's a 'Repositories' section with a 'Repository URL' input containing 'https://github.com/patriciasfo/biblioteca.git' and a 'Credentials' input containing 'patriciasfo/*********'. An 'Avanzado' (Advanced) dropdown is shown. A '+ Add Repository' button is available. The 'Branches to build' section has a 'Branch Specifier' input set to '*/master' and a '+ Add Branch' button. Finally, a 'Navegador del repositorio' (Repository navigator) section is set to '(Auto)'.

Para añadir las credenciales, simplemente seleccionar:

A close-up view of the Jenkins configuration screen for the 'Git' repository. It shows the same fields as the previous screenshot: 'Repository URL' with 'https://github.com/patriciasfo/biblioteca.git', 'Credentials' with 'patriciasfo/*********', and the 'Add' button highlighted with a yellow border. The Jenkins logo is visible in the bottom right corner of the window.

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

Username with password

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

Username ?
patrickasfo

Treat username as secret ?

Password ?
.....

ID ?

Description ?

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

+ Añadir un nuevo paso

Filter

- Ejecutar Ant
- Ejecutar linea de comandos (shell)
- Ejecutar tareas 'maven' de nivel superior
- Ejecutar un comando de Windows
- Invoke Gradle script
- Run with timeout
- Set build status to "pending" on GitHub commit

sending notifications, archiving artifacts, or triggering other jobs.

Jenkins / Tarea maven / Configuration

Configure

Abortar la ejecución si se atasca
 Add timestamps to the Console Output
 Inspect build log for published build scans
 With Ant ?

General
 Configurar el origen del código fuente
 Triggers
 Environment
 Build Steps
 Acciones para ejecutar después.

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Ejecutar tareas 'maven' de nivel superior

Version de Maven
Maven3.9.11

Goles
clean

Avanzado ▾

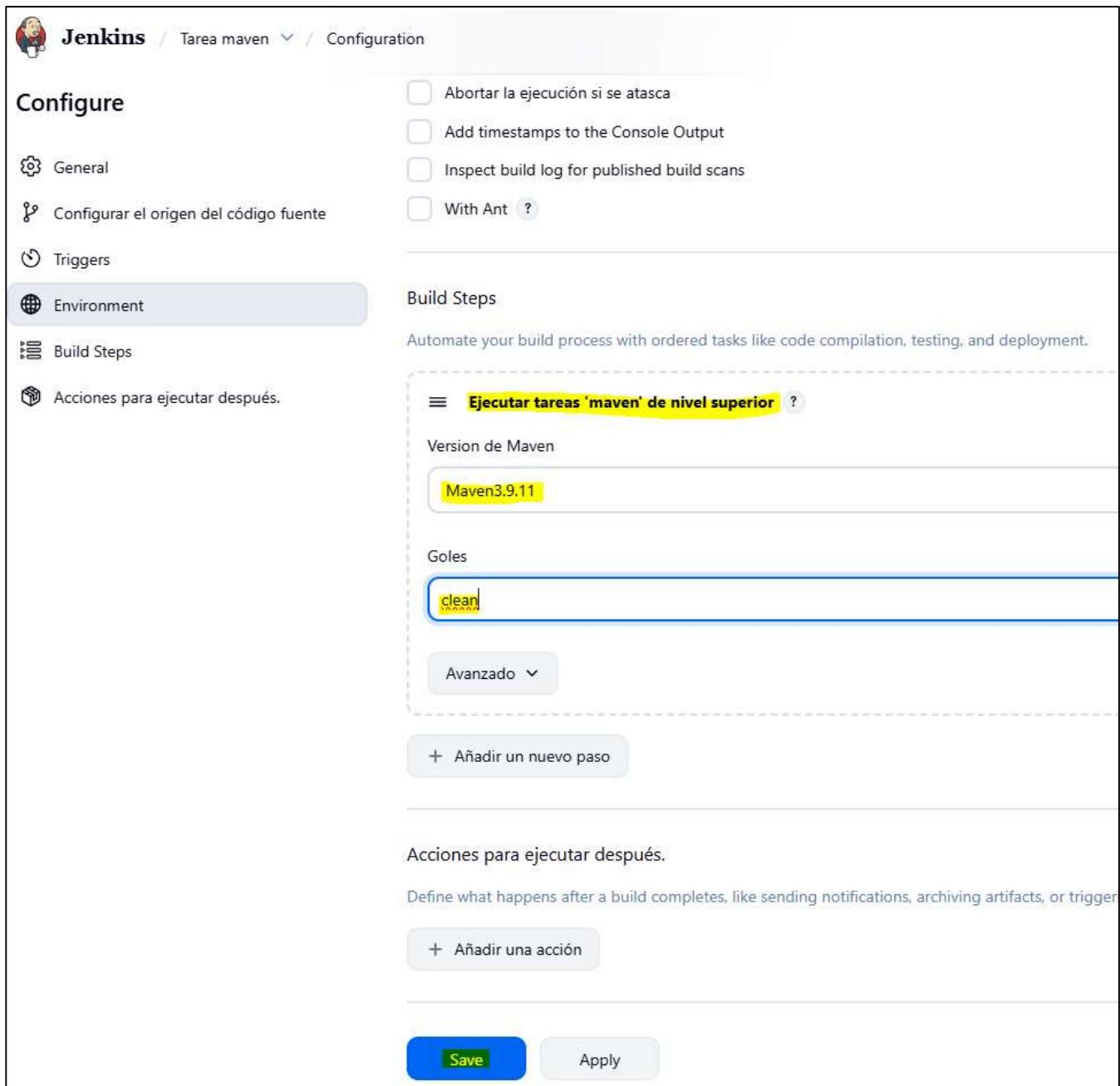
+ Añadir un nuevo paso

Acciones para ejecutar después.

Define what happens after a build completes, like sending notifications, archiving artifacts, or trigger

+ Añadir una acción

Save **Apply**



Para poder escoger la versión de Maven 3.9.11, debemos ir a la configuración de las **Tools** de Jenkis e instalarla:



Administrar Jenkins

Parece que la configuración de proxy inverso está mal

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

System Configuration



System

Configurar variables globales y rutas.



Tools

Configure tools, their locations and automatic installers.



Jenkins

/ Administrar Jenkins ▾ / Tools

instalaciones de Maven

instalaciones de Maven ^

Edited

+ Añadir Maven

☰ Maven

Nombre

Maven3.9.11

Instalar automáticamente ?

☰ Instalar desde Apache

Versión

3.9.11

+ Añadir un instalador

+ Añadir Maven

Save

Apply

Aprovechamos para configuración también la JDK y el GIT:

The screenshot shows the Jenkins interface for managing Java Development Kits (JDKs). The top navigation bar includes the Jenkins logo, 'Administrar Jenkins', and 'Tools'. Below this, the main title is 'instalaciones de JDK'. A sub-header 'instalaciones de JDK ^' has an 'Edited' status indicator. A large button '+ Añadir JDK' is visible. A dashed-line box contains the configuration for a specific JDK entry. Inside this box, the section title is '≡ JDK'. It includes fields for 'Nombre' (containing 'jdk-21') and 'JAVA_HOME' (containing 'C:\Program Files\Java\jdk-21'). There is also a checkbox labeled 'Instalar automáticamente' with a question mark icon. Another '+ Añadir JDK' button is located at the bottom of the configuration area.

The screenshot shows the Jenkins 'Tools' configuration page specifically for 'Git installations'. A single 'Git' entry is listed with the name 'Default'. The 'Path to Git executable' is set to 'C:\Program Files\Git\bin\git.exe'. There is an unchecked checkbox for 'Instalar automáticamente'. A 'Add Git' button is visible at the bottom.

Una vez todo configurado correctamente, ejecutamos la tarea:

The screenshot shows the Jenkins 'Tarea maven' configuration page. The 'Estado Actual' tab is selected, showing the status of the build. The 'Construir ahora' button is highlighted. The 'Zona de Trabajo' section indicates it runs 'Ejecuta el comando de maven clean sobre un proyecto en el repositorio github'. The 'Enlaces permanentes' section lists several URLs related to the build history. The 'Builds' section shows a single build entry for '#1' from today at 19:54.

3. PIPELINE

Lo primero que debemos hacer es crear un **fichero Jenkinsfile vacío en la raíz del proyecto**.

Ahora, creamos la tarea en jenkins:

A screenshot of the "Nuevo Tarea" creation dialog. At the top, it says "Jenkins / Todo / Nuevo Tarea". The title is "Nuevo Tarea". It asks "Enter an item name" with the input field containing "Mi primer pipeline". It then asks "Select an item type" and lists several options: "Crear un proyecto de estilo libre" (description: "Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications."), "Crear un proyecto maven" (description: "Ejecuta un proyecto maven. Jenkins es capaz de aprovechar la configuración presente en los ficheros POM, reduciendo drásticamente la configuración."), "Pipeline" (description: "Gestiona actividades de larga duración que pueden abarcar varios agentes de construcción. Apropiado para construir pipelines (conocidas anteriormente como workflows) y/o para la organización de actividades complejas que no se pueden articular fácilmente con tareas de tipo freestyle."), "Crear un proyecto multi-getConfiguración" (description: "Adecuado para proyectos que requieran un gran número de configuraciones diferentes, como testear en múltiples entornos, ejecutar sobre plataformas concretas, etc."), and "Folder" (description: "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."). A blue "OK" button is at the bottom.

Jenkins / Mi primer pipeline / Configuration

Configure

General

Enabled

Descripción

Este pipeline mostrará un mensaje (tarea de comandos de Windows), hará el checkout de un proyecto en github y ejecutará la tarea de maven clean (tarea de maven)

Plain text [Visualizar](#)

Desechar ejecuciones antiguas ?

Do not allow concurrent builds

Do not allow the pipeline to resume if the controller restarts

Esta ejecución debe parametrizarse ?

GitHub project

Pipeline speed/durability override ?

Preserve stashes from completed builds ?

Throttle builds ?

Jenkins / Mi primer pipeline / Configuration

Configure

Pipeline

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

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?
`https://github.com/patriciasfo/biblioteca.git`

Credentials ?
`patriciasfo/*****`

Avanzado ▾

Add Repository

Branches to build ?

The screenshot shows the Jenkins Pipeline configuration page for a job named "Mi primer pipeline". The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area is titled "Configure". It includes sections for "Branches to build" (with a "Branch Specifier" set to "/master"), "Navegador del repositorio" (set to "(Auto)"), "Additional Behaviours" (with an "Añadir" button), "Script Path" (set to "Jenkinsfile"), and a checked "Lightweight checkout" option. Below these is a "Pipeline Syntax" section with a "Pipeline" example. At the bottom are "Save" and "Apply" buttons.

Antes de construir, tenemos el Jenkisfile vacío, así que ir definiendo su estructura y los diferentes **stages** con las tareas a realizar:

- Se debe empezar con **pipeline {}**.
- Lo primero es definir el **agent**: indica el entorno donde se ejecuta el pipeline. Puede ser un contenedor Docker, un agente en la nube, etc. En nuestro caso, no definiremos ninguno en concreto, por lo tanto, ponemos **any**.
- A continuación, indicamos las **tools** configuradas: en nuestro caso el **jdk** y **maven**. Entre comillas simples debemos indicar el nombre que configuramos en el jenkins.

- Por último, definimos los **stages**, es decir, la lista de actividades a realizar. Cada actividad será un **stage** al que le podemos indicar un **nombre** y los **steps** (pasos) a seguir. Si alguno de los pasos requiere alguna acción a posteriori (**post**), podemos indicarlas, tanto si el paso se realizó con éxito (**success**) o con fracaso (**failure**) o independientemente del resultado (**always**).

```

pipeline {
    agent any

    tools {
        jdk 'jdk-21'
        maven 'Maven3.9.11'
    }

    stages {

        stage('Show messages'){
            steps {
                bat 'echo "Primer stage del pipeline"'
            }
        }

        stage('Checkout proyecto') {
            steps {
                git branch: 'master',
                    url: 'https://github.com/patriciasfo/biblioteca.git'
            }
        }

        stage('Maven clean') {
            steps {
                bat 'mvn clean compile'
            }
        }

        stage('Package') {
            steps {
                bat 'mvn package'
            }
            post {
                success {
                    archiveArtifacts artifacts: 'target/*.jar', fingerprint: true
                }
            }
        }
    }
}

```