

IES JACARANDÁ



**Despliegue de
Aplicaciones Web**

**Despliegue de un proyecto
en Jenkins**

Por: Chisela Marien Colás Gil

2 DAW

Introducción

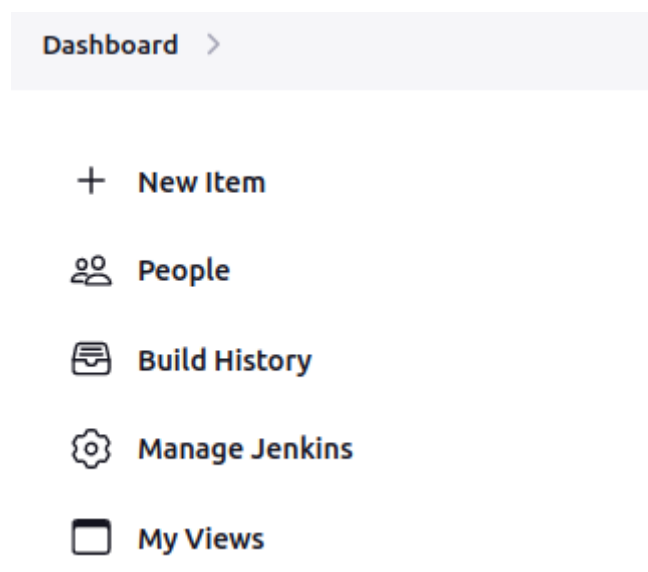
A partir de la configuración básica de Jenkins que vimos en el manual anterior vamos a desplegar un proyecto java a partir de un repositorio en Github.

Pasos previos

Antes de continuar, debemos añadir algunas cosas adicionales a la configuración básica en Jenkins:

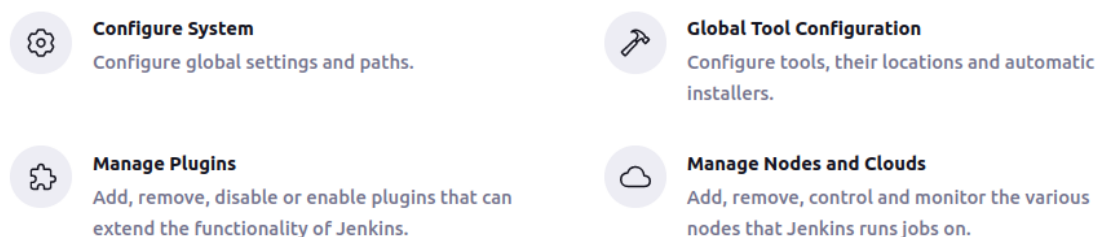
- [Instalación del plugin de Maven](#)

Para instalar el plugin de Maven primero debemos ir a la pestaña “Manage Jenkins”.



Después, a la sección “Manage Plugins”.

System Configuration



Ahora accedemos a la parte de “Available plugins”, tecleamos en la barra de búsqueda “maven”, seleccionamos el plugin “Maven integration” y nos aparecerán dos botones para instalarlo; elegimos instalar sin reiniciar.

Dashboard > Manage Jenkins > Plugin Manager

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Plugins


Q maven


Install	Name ↓	Released
<input checked="" type="checkbox"/>	<div>Maven Integration 3.21</div> <div>Build Tools</div> <div>This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTS as well as the automated configuration of various Jenkins publishers such as Junit.</div>	10 days ago


- Configuración del entorno de trabajo


Para configurar el entorno de trabajo, volvemos a la sección de “Manage Jenkins” y seleccionamos “Global Tool Configuration”.

System Configuration

**Configure System**
Configure global settings and paths.

**Global Tool Configuration**
Configure tools, their locations and automatic installers.

**Manage Plugins**
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

**Manage Nodes and Clouds**
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Al final de esta página encontraremos la sección de Maven, en ella le damos a “Add Maven”, le damos un nombre y seleccionamos la opción de instalar automáticamente.

Dashboard > Manage Jenkins > Global Tool Configuration

Maven

Name

maven_jenkins

☒ Install automatically ?

Install from Apache


Version

3.9.0

Add Installer ▾

Creación del proyecto

Una vez acabados estos sencillos pasos previos, vamos a crear nuestro proyecto. Lo primero es seleccionar “New Item” en nuestro Dashboard, en la siguiente ventana le daremos un nombre y seleccionaremos “Maven project”.



Enter an item name

soccer

» Required field

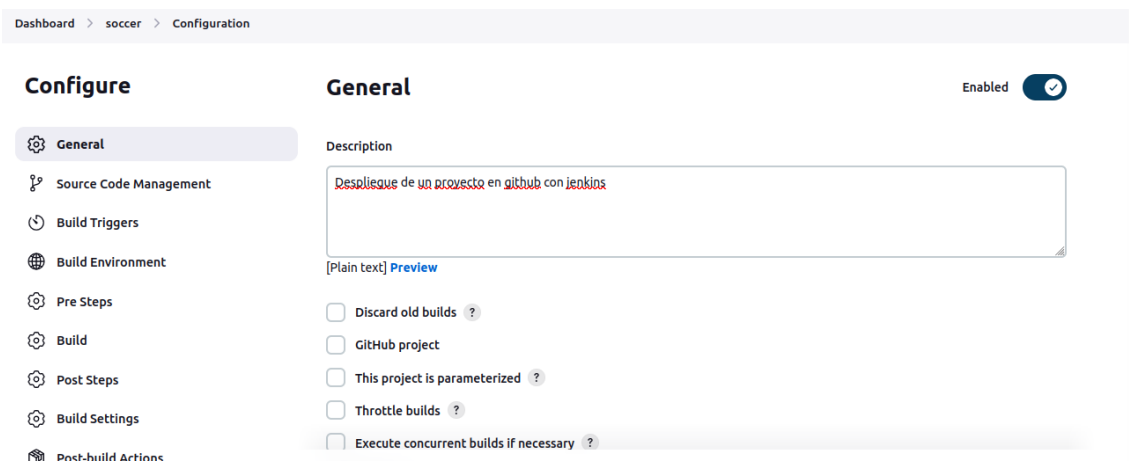
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

OK

Ahora configuraremos los elementos necesarios para que el despliegue funcione. De manera opcional darle una descripción al proyecto.



Dashboard > soccer > Configuration

Configure

General Enabled

Description

Despliegue de un proyecto en github con jenkins

[Plain text] Preview

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

Como el código de nuestro proyecto proviene de un repositorio online, seleccionamos la opción de Git en “Source Code Management” e introducimos la url <https://github.com/webipedia/soccer>. Como es un proyecto público cuya rama principal es “master” no tenemos que añadir nada más.

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/webipedia/soccer

Credentials ?

- none -

+ Add

Advanced ▾

La última configuración que vamos a tocar está relacionada con la integración continua. Aquí podemos configurar una serie de opciones para que se construya el proyecto de diversas formas: nosotros vamos a indicar por ejemplo que recoja el código del repositorio cada minuto en busca de cambios.

Build Triggers

☒ Build whenever a SNAPSHOT dependency is built ?

☐ Schedule build when some upstream has no successful builds ?

☐ Trigger builds remotely (e.g., from scripts) ?

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☒ Poll SCM ?

Schedule ?

⚠ Do you really mean "every minute" when you say "*****"? Perhaps you meant "H*****" to poll once per hour

Would last have run at Saturday, March 11, 2023 at 2:33:18 PM Coordinated Universal Time; would next run at Saturday, March 11, 2023 at 2:33:18 PM Coordinated Universal Time.

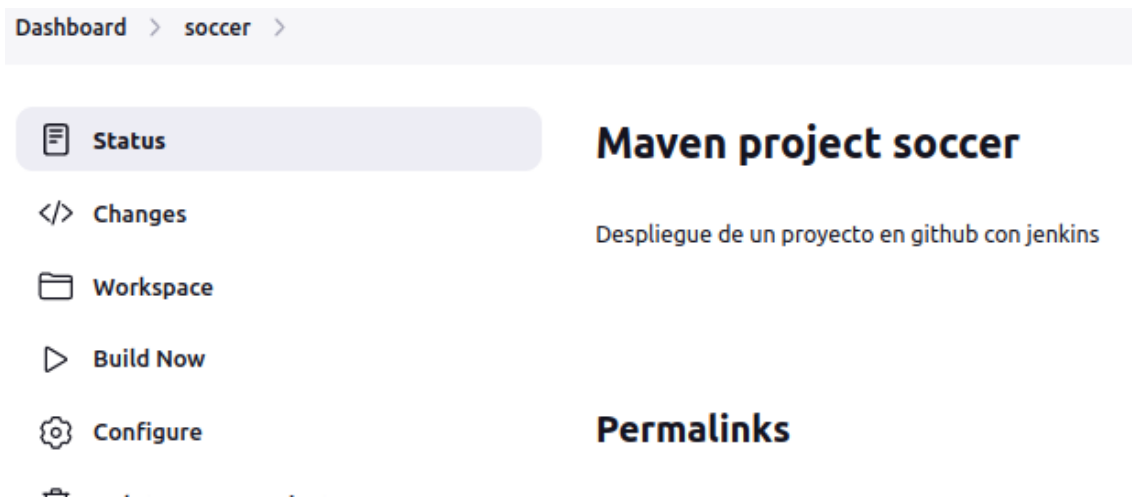
Para que la construcción del proyecto funcione tenemos que asegurarnos que aparezca el fichero pom.xml en la sección "Build". Después guardamos la configuración.

Build

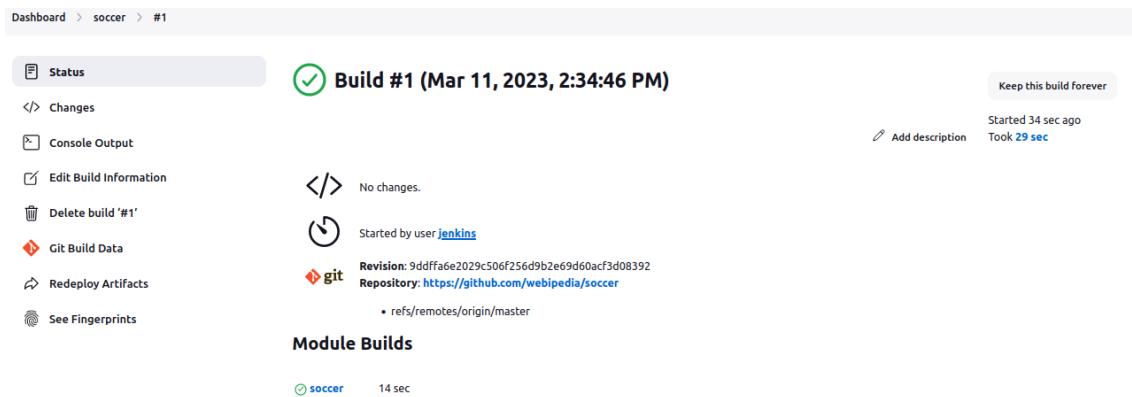
Root POM ?

pom.xml

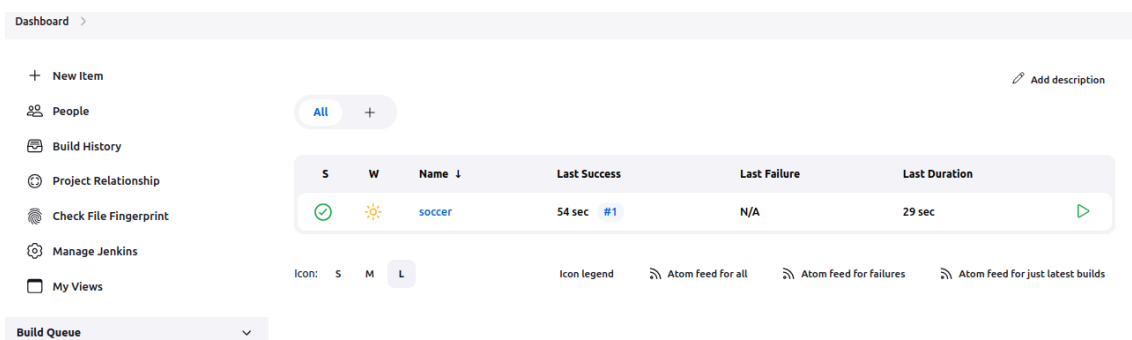
El último paso es construir el proyecto. Para ello seleccionamos la opción “Build Now”.



Si todo va bien veremos en pantalla algo parecido a lo siguiente:



Y si volvemos a nuestro Dashboard podremos ver información resumida sobre el estado de nuestro proyecto.



Referencias web

- [Tutorial utilizado](#)