



Certified Tech Developer

The Ultimate Degree

Infraestructura II

Actividad obligatoria e individual

Dificultad: Media

Continuous Integration con Jenkinsfile - Parte 1

Para esta práctica vamos a tener como base el video sobre el proceso de Continuous Integration con Jenkins. En esa oportunidad realizamos el proceso sin repositorio de código en un solo paso. Es decir, no era un pipeline, sino una tarea dedicada a compilar aplicaciones Java. **¡Mejoremos nuestra automatización!**

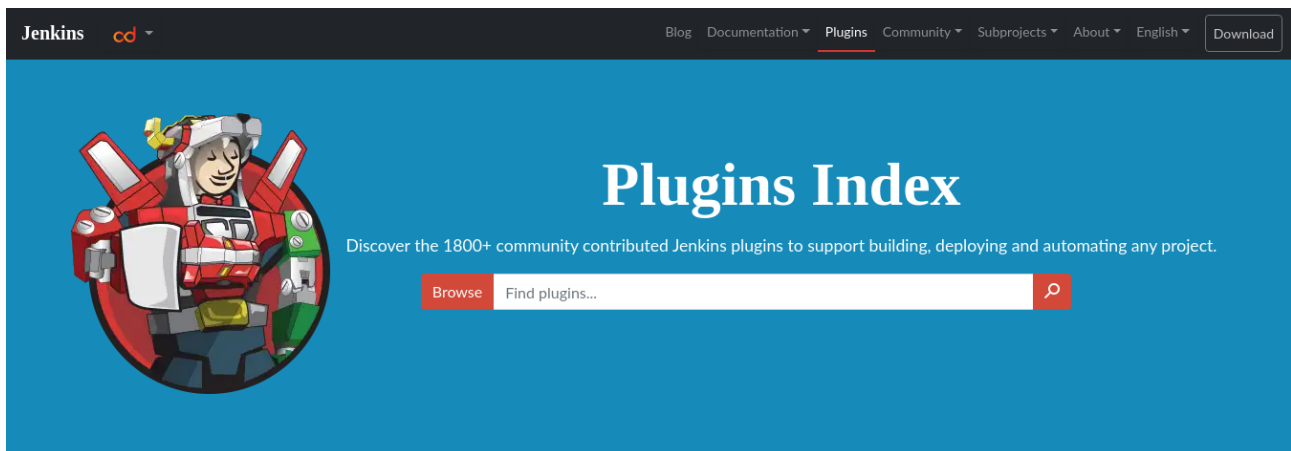
Plugins

En Jenkins vamos a poder ejecutar tareas para la gran mayoría de las tecnologías del mercado. Incluso, vamos a poder ejecutar tareas integrándonos con otras herramientas. Todo esto gracias a los plugins.

Búsqueda de plugins

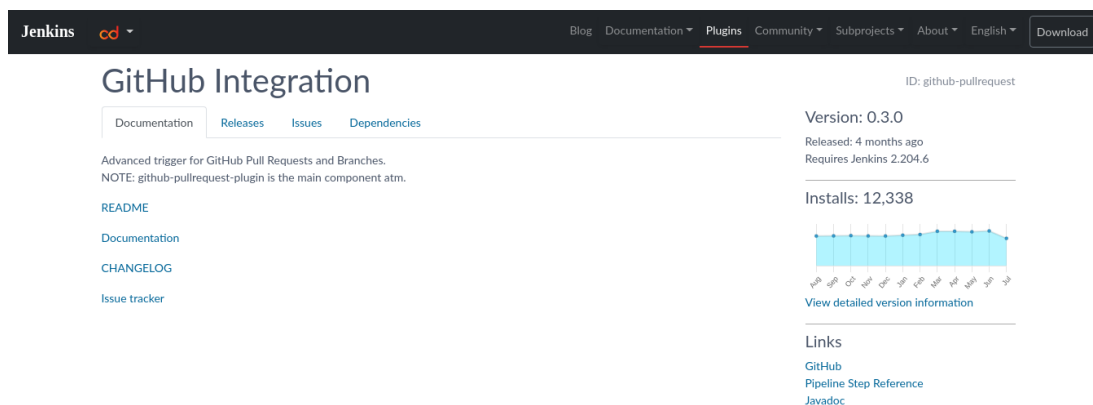
Todos los plugins se encuentran centralizados en el siguiente enlace: [Jenkins Plugins](#)

En el área de texto de la página principal vamos a poder realizar la búsqueda de los plugins que necesitemos y podremos utilizarlos dentro de nuestros Jenkinsfiles de la forma más adecuada para nuestro propósito.



Tenemos que leer la documentación del plugin, revisar que se encuentre actualizado, y modificar los parámetros del plugin para enviar y recibir la información que necesitamos.

Analizando el plugin “GitHub Integration”, solo a modo de ejemplo, tenemos que reconocer algunos campos para identificar si nos ayuda a realizar nuestro propósito.



Versionado de plugins

Hay que visualizar si el plugin fue recientemente actualizado. Recordemos que los plugins son mantenidos por la comunidad, por lo que su desarrollo puede ser abandonado por diversas razones. Esto nos impacta negativamente, ya que podríamos necesitar un plugin que no está actualizado con respecto a las tecnologías que se integra. También es importante tener en cuenta la versión de Jenkins que hay que tener instalada:



Version: 0.3.0

Released: 4 months ago

Requires Jenkins 2.204.6

Dependencias de plugins

Como cualquier desarrollo, los plugins también utilizan código ya hecho para partir de ese punto y crear una solución que resuelve un problema. Cada desarrollo utiliza otros plugins como base, y por lo tanto, son dependencias que deben instalarse previamente (no se preocupen, se instalan automáticamente).

Es necesario que visualicemos ese punto antes de instalar un plugin, ya que si una dependencia coincide con alguna ya instalada en nuestro Jenkins, pero varía su versión, puede fallarnos el funcionamiento de otro plugin ya instalado.

GitHub Integration

[Documentation](#)

[Releases](#)

[Issues](#)

[Dependencies](#)

Dependencies

Required

GitHub ≥ 1.28.1

Pipeline: API ≥ 2.23.1

Pipeline: Multibranch ≥ 2.16

Pipeline: Step API ≥ 2.13

Branch API ≥ 2.0.16

Folders ≥ 6.2.1

Credentials ≥ 2.1.16

Git client ≥ 2.6.0

Git ≥ 3.6.3

GitHub API ≥ 1.86

Mailer ≥ 1.18

Matrix Project ≥ 1.7.1

SCM API ≥ 2.2.3

Script Security ≥ 1.34

SSH Credentials ≥ 1.13

JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI) ≥ 1.2.1

ID de un plugins

Los plugins también tienen identificadores únicos. De esta manera, una vez elegidos, vamos a poder instalarlos buscando a través de ese identificador e instalarlos fácilmente desde la interfaz gráfica de Jenkins.

ID: github-pullrequest

Instalamos plugins y los utilizamos en pipelines

¡Manos a la obra! Ahora que conocemos lo básico sobre los plugins, ejecutamos nuestra instancia de Jenkins e instalamos los plugins necesarios para realizar la compilación de una aplicación Java bajo un esquema de Continuous Integration.

Busquemos dentro de la interfaz gráfica lo siguiente:

- Recorremos las opciones de configuración de Jenkins e identificamos dónde se administran los plugins.
- Descubrimos las opciones que tenemos y cómo buscar los plugins:
 - Maven Integration.
 - GitHub.
 - GitHub Integration.
- Vamos a instalar cada plugin.
- Identificamos dentro de la configuración de Jenkins cómo guardar credenciales. Para esta práctica, de nuestra cuenta de GitHub.
- Creamos un repositorio de código para versionar nuestro Jenkinsfile y una aplicación de ejemplo. Para completar esta práctica, está bien subir el Jenkinsfile que solo ejecute "echo".

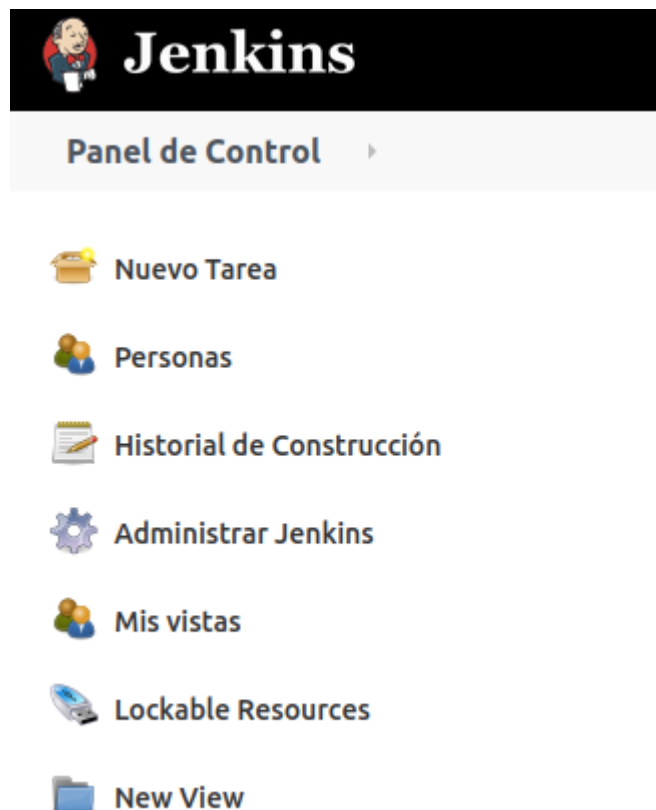
En la siguiente página se encuentra la resolución. Continúa únicamente para realizar una autoevaluación.

Resolución

- **Recorremos las opciones de configuración de Jenkins e identificamos donde se administran los plugins.**

Realizamos la instalación de plugins, utilizando la interfaz gráfica, dentro del siguiente apartado de Jenkins.

En la pantalla principal, cliqueamos dentro “Administrar Jenkins” en el menú de la izquierda



Vamos a visualizar una sección dedicada únicamente a plugins.



Administrar Jenkins

Hay una nueva versión de Jenkins disponible (2.303.1). [descargar](#) ([listado de cambios](#)).

System Configuration



Configurar el Sistema

Configurar variables globales y rutas.



Global Tool Configuration

Configure tools, their locations and automatic installers.



Administrar Plugins

Añadir, borrar, desactivar y activar plugins que extienden la funcionalidad de Jenkins.

- **Descubrimos las opciones que tenemos y cómo buscar los plugins:**
 - **Maven Integration.**
 - **GitHub.**
 - **GitHub Integration.**

El gestor de plugins consta de cuatro secciones:

- Actualizar los instalados.
- Disponibles para descarga.
- El listado de todos los que tenemos instalados.
- Una sección de avanzado para subir plugins manualmente y realizar configuraciones de red.

Actualizar	Disponibile	Instalado	Avanzado
Instalar	Nombre ↓		

- **Vamos a instalar cada plugin.**

Para nuestra práctica, vamos a clicar en “Disponibile” y buscar los plugins que necesitamos.



Vamos a seleccionar cada uno y elegir la opción “Instalar sin reiniciar”. Esto es para instalar varios plugins y que el reinicio –para que impacte el cambio– sea al final de todas las instalaciones.

Actualizar

Disponible

Instalado

Avanzado

Install	Name	Version	Released
<input checked="" type="checkbox"/>	GitHub Integration emailtext Plugins lanzadores de tareas GitHub Integration Plugin for Jenkins	0.3.0	4 Mes 8 días ago
<input type="checkbox"/>	DotCi Github integration, build management through .ci.yml, environment management though docker.	2.40.00	3 Año 6 Mes ago

Install without restart

Download now and install after restart

Update information obtained: 1 Hor 1 Min ago

Comprobar ahora

Nos redirige a una pantalla similar a esta, donde visualizamos los chequeos que realiza y confirmamos que instaló las dependencias con el plugin.

Instalando Plugins/Actualizaciones

Preparación

- Probando conectividad con Internet
- Probando conectividad con jenkins-ci.org
- Correcto

JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI) Actualizado

GitHub Integration Actualizado

Loading plugin extensions Success

Una vez que finalicemos, vamos a clicar en la última opción para reiniciar Jenkins y tener disponible todas las funcionalidades de los plugins instalados.



Instalando Plugins/Actualizaciones

Preparación

- Probando conectividad con Internet
- Probando conectividad con jenkins-ci.org
- Correcto

JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI)	✓ Actualizado
GitHub Integration	✓ Actualizado
Loading plugin extensions	✓ Success
Javadoc	✓ Actualizado
Maven Integration	✓ Actualizado
Loading plugin extensions	✓ Success
GitHub Authentication	✓ Actualizado
Loading plugin extensions	✓ Success
GitHub	⚠ Descarga correcta. Se activará en el próximo arranque.

➡ [Volver al inicio de la página](#)

(puedes empezar a utilizar los plugins instalados inmediatamente)

➡ ☐ Reiniciar Jenkins cuando se complete la instalación y no haya trabajos corriendo

- **Identificamos dentro de la configuración de Jenkins como guardar credenciales. En nuestro caso, de nuestra cuenta de GitHub.**

Dentro de la configuración de Jenkins vamos al apartado de seguridad y clic en “Manage Credentials” para administrar las credenciales que necesitamos para nuestros pipelines.

Security



Configuración global de la seguridad

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



Manage Credentials

Configure credentials



Configure Credential Providers

Configure the credential providers and types



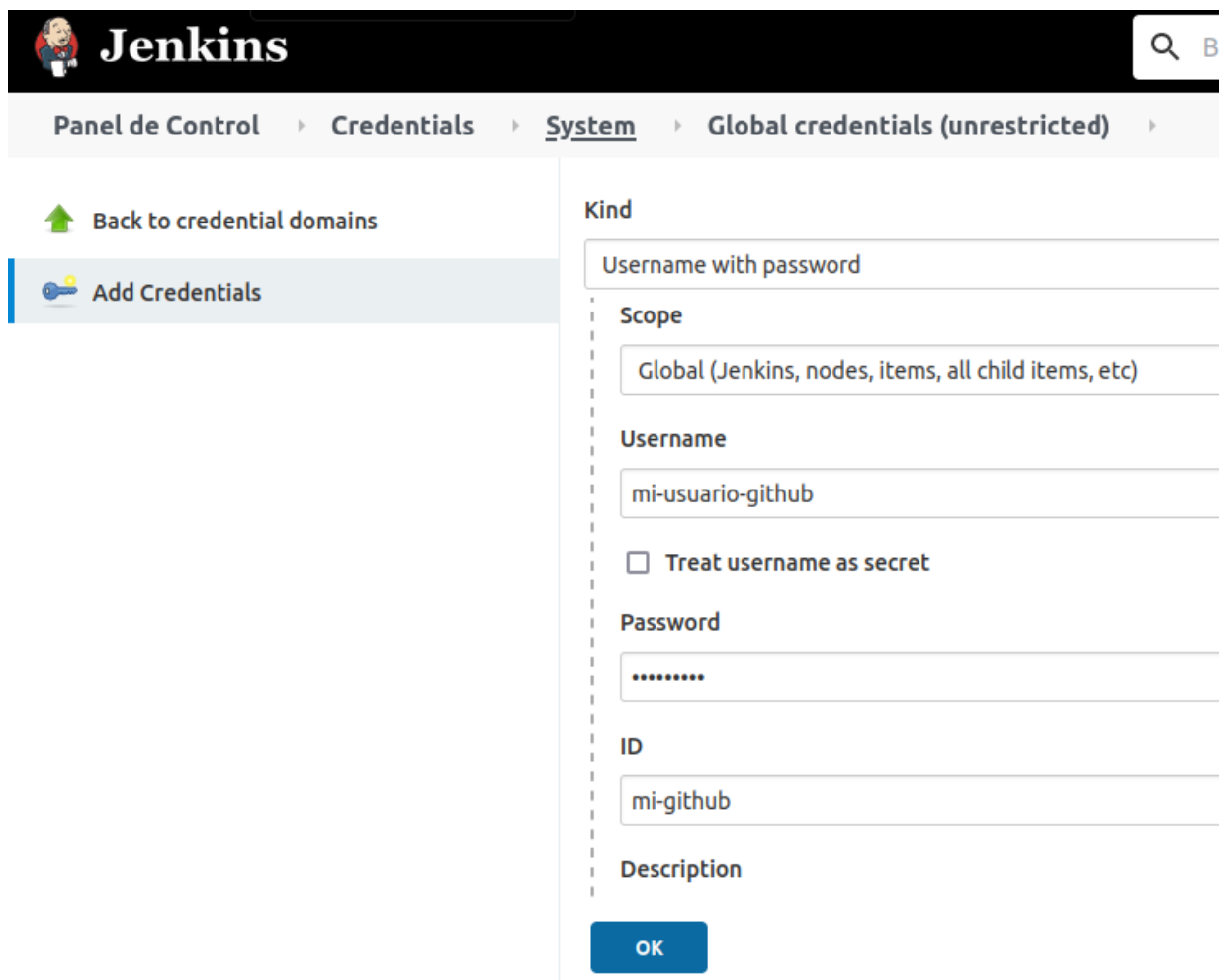
Gestión de usuarios

Crear/borrar/editar usuarios que puedan utilizar Jenkins



En nuestro caso vamos a guardarlas con alcance global. Esta es la mejor opción para que nuestras prácticas sean más sencillas al trabajar con una cuenta personal y para uso académico.

Tenemos varias opciones. Para nuestras credenciales necesitamos que sea por "Username with password". En el apartado ID vamos a colocar un nombre para nuestras credenciales, que luego nos ayudará a identificarlas al crear los pipelines. Este último es un identificador único que no se puede repetir con otras credenciales.



The screenshot shows the Jenkins web interface. At the top, there's a black header with the Jenkins logo and name. Below it, a navigation bar contains links: "Panel de Control", "Credentials", "System", and "Global credentials (unrestricted)". The left sidebar has two buttons: "Back to credential domains" and "Add Credentials". The main content area is titled "Kind" and shows a form for creating a new credential. The form fields are: "Kind" (set to "Username with password"), "Scope" (set to "Global (Jenkins, nodes, items, all child items, etc)"), "Username" (set to "mi-usuario-github"), "Treat username as secret" (unchecked), "Password" (masked with dots), "ID" (set to "mi-github"), and "Description" (empty). An "OK" button is at the bottom.

Jenkins

Panel de Control > Credentials > System > Global credentials (unrestricted)

Back to credential domains

Add Credentials

Kind

Username with password

Scope

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

Username

mi-usuario-github

☐ Treat username as secret

Password

.....

ID

mi-github

Description

OK



- **Creamos un repositorio de código para versionar nuestro Jenkinsfile y una aplicación de ejemplo. Para completar esta práctica, está bien subir el Jenkinsfile que solo ejecute "echo".**

En nuestra cuenta de GitHub, creamos un repositorio para subir únicamente un Jenkinsfile de ejemplo. Si nunca creaste un repositorio, visitá esta guía en español:

[Crear un repositorio](#)

El repositorio que utilizaremos es el oficial de la materia. Vamos a crear una carpeta con esta práctica. El archivo en cuestión se encuentra en el siguiente enlace:

<https://raw.githubusercontent.com/repoinfradh/Infra2/main/practica-pipelines/Jenkinsfile>

La estructura de este archivo nos acompañará durante varias clases. En las prácticas que siguen, vamos a comenzar reemplazando el primer bloque de "Build".

Ahora que completaste todas las prácticas de esta clase. ¿Qué otros plugins te gustaría probar? ¿Qué sabemos de los plugins instalados? En la documentación oficial, ¿hay información que nos ayude a probarlos?