

**Centro de Enseñanza Técnica Industrial**

**Desarrollo de Software**

**Desarrollo Continuo**

**Jesús Alberto Aréchiga Carrillo**

**22310439 6N**

**Profesor**

**José Francisco Pérez Reyes**

**Junio 2025**

**Guadalajara, Jalisco**

## Objetivo

Automatizar el proceso de integración continua donde:

* Los desarrolladores suben código a un repositorio Git.
* Al hacer un *push*, Jenkins es notificado.
* Jenkins ejecuta automáticamente una compilación o pruebas.

### Requisitos Previos

* Jenkins instalado y funcionando (http://localhost:8080 o en un servidor).
* Plugin Git instalado en Jenkins.
* Repositorio Git local y remoto
* Acceso al servidor Jenkins desde el repositorio

### Crear el Proyecto en Jenkins

1. Ir a Jenkins → *New Item* → elegir **Freestyle project**.
2. Asignar un nombre (por ejemplo: MiProyectoCI).
3. En la sección **Source Code Management**:

* Elegir **Git**.
* Poner la URL del repositorio.
* Si es privado, agregar las credenciales.

1. En la sección **Build Triggers**:

* Activar **Build when a change is pushed to GitHub** (si usas Webhooks).
* O bien **Poll SCM** (para checar periódicamente si hay cambios)

## COMANDOS A UTILIZAR

En la sección **Build**:

* Agregar un paso para compilar, por ejemplo:

**make build**

**Configurar Git Hook (si Jenkins y Git están en la misma máquina)**

Dentro del repositorio local de Git, se puede usar el hook post-commit o post-push para que llame a Jenkins.

**<tu\_repo\_local>/.git/hooks/post-commit**

Contenido del hook (post-commit):

**#!/bin/bash**

**curl -X POST** [**http://localhost:8080/job/MiProyectoCI/build**](http://localhost:8080/job/MiProyectoCI/build)

**Para permisos de autenticacion**

chmod +x .git/hooks/post-commit

**Configurar Webhook en GitHub/GitLab (alternativa más común)**

1. Entra a tu repositorio en GitHub.
2. Ve a **Settings** → **Webhooks** → *Add webhook*.
3. En **Payload URL** pon:

**http://<IP\_o\_dominio\_de\_Jenkins>:8080/github-webhook/**

1. **Content type: application/json.**
2. **Which events: elige Just the push event.**
3. Guardar.

**OJO!!**

**Jenkins debe ser accesible desde Internet o estar en la misma red.**

Token de Jenkins (si es necesario)

En Jenkins:

1. Clic en tu nombre de usuario → *Configure*.
2. En "API Token", generar uno nuevo si no hay.
3. Usar ese token en curl o al configurar el webhook si requiere autenticación.

**Verificar Funcionamiento**

1. Haz un cambio en tu código.
2. Realiza el push:

## BASH

git add .

git commit -m "Cambio de prueba"

git push origin main

### Comandos relevantes

# Inicializar repositorio Git

git init

# Configurar remoto

git remote add origin <https://github.com/usuario/repositorio.git>

# Añadir y hacer commit

git add .

git commit -m "Primer commit"

# Subir cambios

git push -u origin main

Usar webhooks es preferible a hooks locales si estás trabajando en equipo.

Si Jenkins no está en red pública, puedes usar herramientas como ngrok para exponerlo temporalmente.

Asegúrate de tener habilitado el puerto 8080 y de que Jenkins permita peticiones externas.