

# Guía para el Job de Jenkins: Pipeline de Node.js

## Objetivo

Este documento proporciona una guía detallada sobre cómo utilizar el pipeline de Jenkins creado para la prueba de concepto de un proceso CI/CD para una API desarrollada en Node.js. El job está diseñado para construir, probar y ejecutar la aplicación automáticamente cada vez que hay cambios en el repositorio.

## Escenario

Nuestra organización está llevando a cabo una prueba de concepto para establecer un flujo de trabajo de CI/CD utilizando Node.js. Se ha estructurado un job de Jenkins que ejecutará automáticamente el build y las pruebas de la aplicación cada vez que se produzca un `push` o un `pull request` en el repositorio configurado.

## Requisitos

1. **Repositorio:** Crear un proyecto en node js, como por ejemplo el siguiente [nodejs-helloworld-api](#).
2. **Configuración del Webhook** en GitHub: Configurar un webhook en el repositorio que active el job de Jenkins cada vez que se produce un `push` o se crea un `pull request`.
3. **Exponer Jenkins mediante Ngrok:** Utilizar Ngrok para exponer el servicio de Jenkins, permitiendo que GitHub envíe notificaciones a tu servidor Jenkins.
4. **Modificaciones en el Proyecto:** Realizar los cambios necesarios en el proyecto y realiza un `push` o crea un `pull request` para activar automáticamente la ejecución del job.

## Cómo utilizar el job

1. Este job compila y ejecuta pruebas del contenido del repositorio indicado ([nodejs-helloworld-api](#)).
2. Para utilizarlo basta con clonar el proyecto indicado en un repositorio local.
3. Hacer los cambios necesarios en el proyecto y realizar un `push` o crea un `pull request` desde github.com. Cualquiera de estos eventos activará automáticamente la ejecución del job.

## Pipeline de Jenkins

El pipeline realiza las siguientes etapas:

1. **Clonar el Repositorio:** Descarga el código fuente desde el repositorio de GitHub.
2. **Limpiar Instalaciones Previas:** Elimina `node_modules` y `package-lock.json` para asegurarse de que la instalación de dependencias sea limpia.
3. **Instalar Dependencias:** Ejecuta `npm install` para instalar las dependencias necesarias.

4. **Ejecutar Pruebas:** Corre los tests definidos en el proyecto utilizando Jest.

## Configuración del Pipeline

A continuación se muestra la configuración del pipeline en Groovy:

```
pipeline {
    agent any

    tools {
        nodejs 'NodeJS'
    }

    stages {
        stage('Clone Repository') {
            steps {
                git url: 'https://github.com/hernan-fb/testUseForJenkins', branch: 'main'
            }
        }
        stage('Clean Previous Installations') {
            steps {
                sh 'rm -rf node_modules package-lock.json || true'
            }
        }
        stage('Install Dependencies') {
            steps {
                sh 'npm install'
            }
        }
        stage('Run Tests') {
            steps {
                sh 'npm test'
            }
        }
    }
    post {
        success {
            echo 'El proceso se ejecutó correctamente.'
        }
        aborted {
            echo "El proceso fue abortado correctamente"
        }
        failure {
            echo 'Hubo un error durante la ejecución del proceso'
        }
    }
}
```