

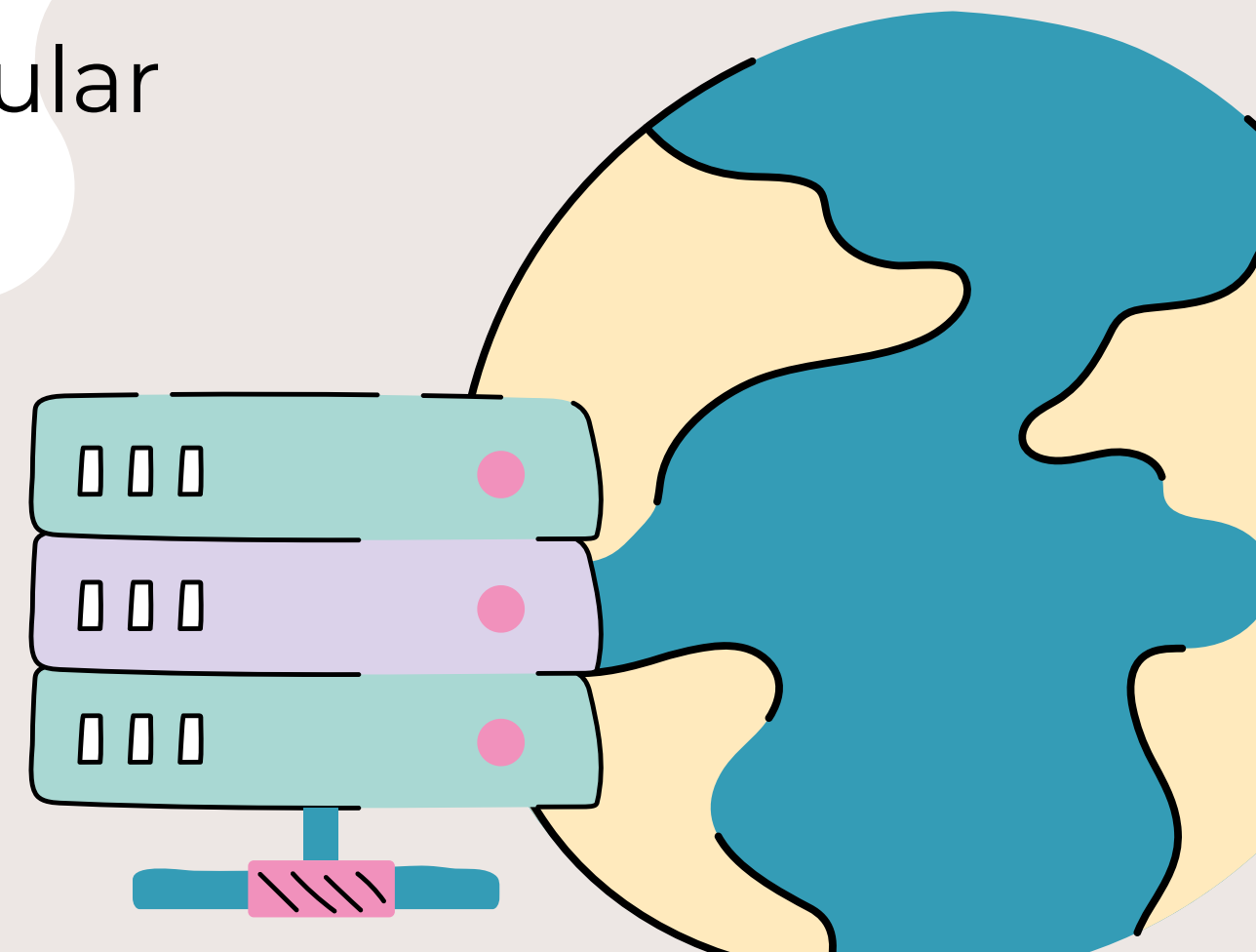


# JENKINS

Una introducción práctica a la herramienta de  
integración continua más popular



Por Juan Duran



# ¿Por qué Jenkins?

Vivimos en un mundo donde el **desarrollo** de **software** necesita ser **rápido, confiable** y **automatizado**. Los equipos de desarrollo ya no pueden permitirse esperar días para desplegar cambios o corregir errores.

Aquí es donde entra **Jenkins**:

Una **herramienta de integración continua (CI)** que permite automatizar procesos de desarrollo, como compilación, pruebas y despliegue.

Con Jenkins puedes:

- **Detectar** errores temprano
- **Acelerar** la entrega de software
- **Aumentar** la eficiencia del equipo



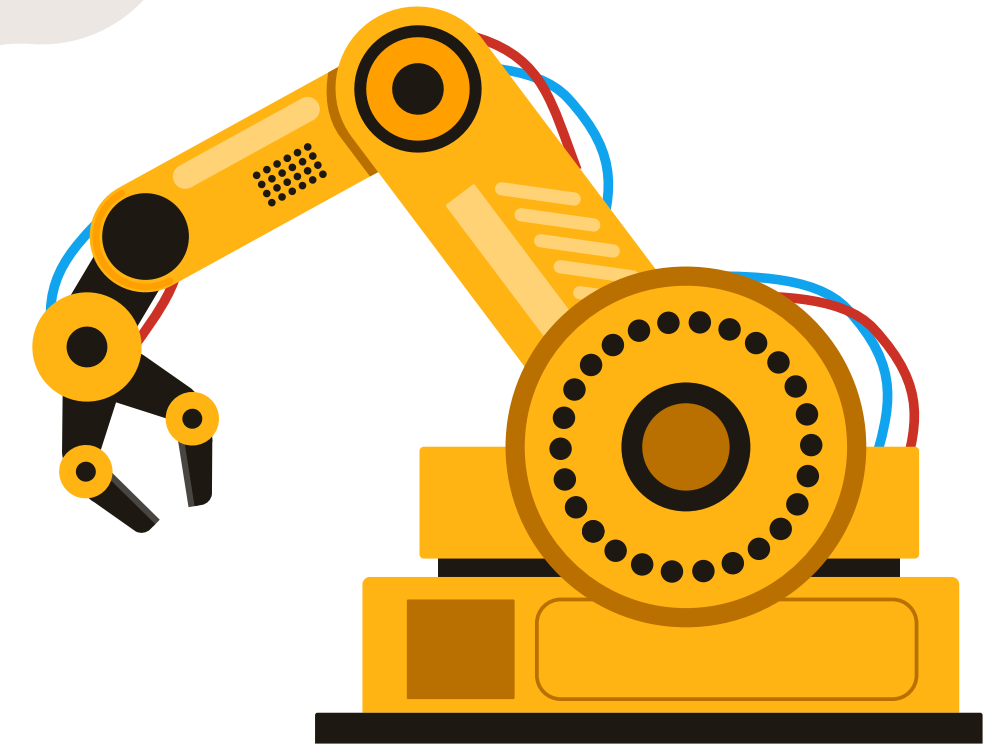
# ¿Qué es Jenkins exactamente?

**Jenkins** es un **servidor** de **automatización** open source, escrito en Java, que permite crear **pipelines** (flujos de trabajo automatizados) para **desarrollar, probar** y **desplegar** tu código.

Es altamente **configurable, extensible** y **compatible** con muchas herramientas, lenguajes y entornos.

Se **integra** fácilmente con Git, Docker, Kubernetes, Maven, Slack, etc.

En resumen: **Jenkins** es el motor que impulsa la **automatización** en proyectos de software modernos.



# ¿Qué es Integración Continua (CI)?

La **Integración Continua** consiste en **integrar cambios** de código frecuentemente (a diario o incluso varias veces al día) en un repositorio compartido, donde se validan **automáticamente** con **pruebas** y **compilaciones**.

## Ventajas:

- ✓ Detectar errores rápido
- ✓ Tener una versión funcional del proyecto en todo momento
- ✓ Menos conflictos entre desarrolladores
- ✓ Mayor confianza en el código

**Jenkins** permite **automatizar** todo este proceso con cada cambio realizado en el repositorio.



# Jenkins en acción: ¿Cómo funciona?

**Jenkins trabaja** a través de **jobs** o tareas que se ejecutan de forma **automática** cuando se produce un evento, como un push en Git.

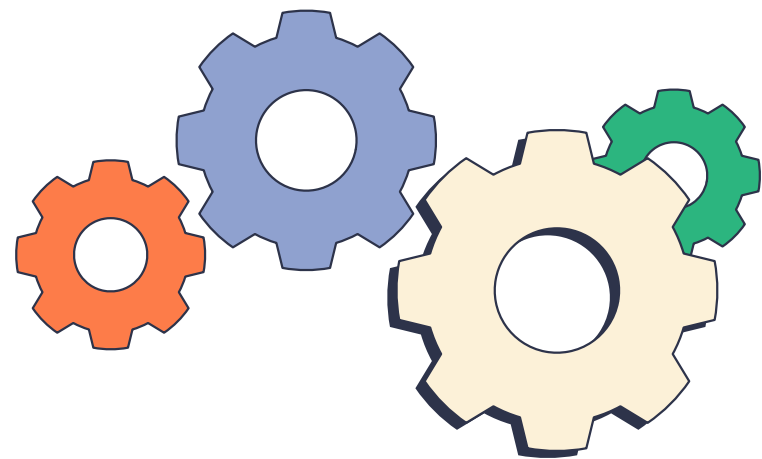
**Funcionamiento** básico:

- 1.El desarrollador sube código a GitHub/GitLab
- 2.Jenkins detecta el cambio
- 3.Compila el código
- 4.Ejecuta pruebas
- 5.Si todo está bien, lo despliega

**Esto se conoce como Pipeline CI/CD: Integración + Entrega continua.**



# ¿Qué es un Pipeline en Jenkins?



Un **pipeline** es una **secuencia** de **etapas** o pasos **automatizados** para construir, probar y desplegar código.

En Jenkins puedes definir estos pipelines mediante un **archivo** llamado **Jenkinsfile**, que contiene **instrucciones** como:

```
pipeline {
  agent any
  stages {
    stage('Build') {
      steps { echo 'Compilando...' }
    }
    stage('Test') {
      steps { echo 'Ejecutando pruebas...' }
    }
    stage('Deploy') {
      steps { echo 'Desplegando aplicación' }
    }
  }
}
```

# Plugins: El poder de Jenkins

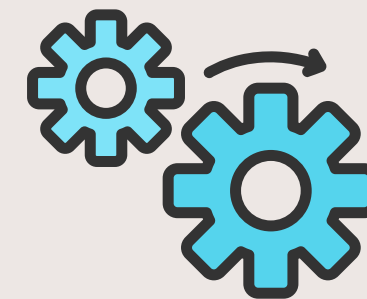


Una de las grandes **ventajas** de **Jenkins** es su **ecosistema de plugins**.

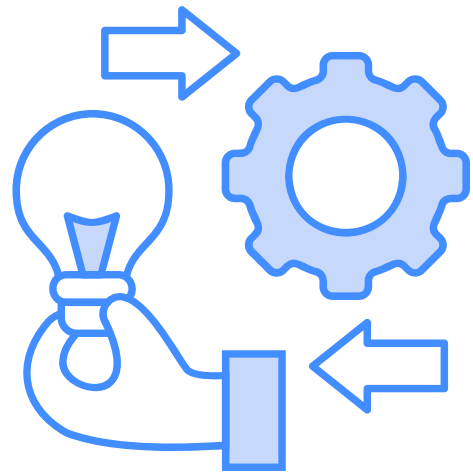
Tiene más de **1800 plugins** disponibles para integrarse con todo tipo de herramientas:

- Git, GitHub, Bitbucket
- Docker y Kubernetes
- SonarQube (análisis de código)
- Slack (notificaciones)
- JUnit, Selenium (pruebas)
- AWS, Azure, GCP

**Esto lo convierte en una plataforma increíblemente versátil.**



# Ventajas y desventajas de Jenkins



## Ventajas:

- ✓ Es gratuito y de código abierto
- ✓ Gran comunidad y documentación
- ✓ Compatible con casi todo
- ✓ Muy flexible y personalizable
- ✓ Permite escalar con facilidad

## Desventajas:

- ✗ La interfaz puede resultar anticuada
- ✗ Requiere configuración inicial manual
- ✗ Hay que mantener los plugins actualizados
- ✗ Curva de aprendizaje si es tu primera herramienta de CI/CD





# Casos de uso reales



**Jenkins** se usa en miles de empresas de todo tipo para automatizar tareas como:

- ◆ **Despliegue** de microservicios
- ◆ **Testing** automático en apps móviles y web
- ◆ **Creación** de **contenedores** Docker
- ◆ **Actualización** de **infraestructuras** en la nube
- ◆ **Notificaciones** de fallos en Slack o Teams
- ◆ **Generación** de **informes** automáticos

Su uso ahorra horas de trabajo manual y reduce errores humanos.



# Conclusión



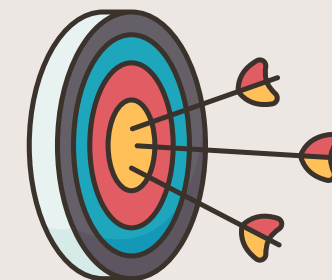
**Jenkins** es una **herramienta** clave en el mundo del **desarrollo** moderno. Permite **automatizar** desde tareas simples hasta flujos de trabajo complejos.

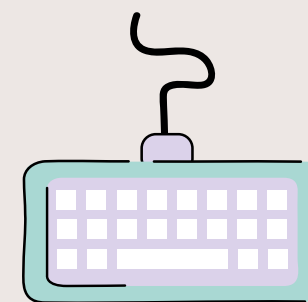
Si estás aprendiendo **desarrollo**, **DevOps** o trabajando con **datos**, conocer Jenkins te abre muchas puertas.

**Próximos pasos** que puedes dar:

- ◆ Instalar Jenkins en local (o usarlo en la nube)
- ◆ Conectarlo a tu repositorio Git
- ◆ Crear tu primer pipeline
- ◆ Explorar plugins útiles
- ◆ Integrarlo con herramientas como Docker o Slack

**La automatización es el futuro, y Jenkins es una excelente puerta de entrada.**





# Gracias



Por Juan Duran

“Coding, Gaming and Leveling Up”