**JENKINS**



**Nombre:** Javier Villarreal Hinojo

**Módulo:** Despliegue de aplicaciones Web

**Fecha:** 03-02-2023

**ÍNDICE**

[**QUÉ ES JENKINS**](#_s1a8efjgvr34) **3**

[**¿QUÉ SE PUEDE HACER CON JENKINS?**](#_eg60y3h8st8l) **3**

[**¿CÓMO FUNCIONA JENKINS?**](#_vcmn5flvpgo0) **3**

[**VENTAJAS DE USAR JENKINS**](#_xqwq6ef1ft0s) **4**

[**QUÉ ES LA INTEGRACIÓN CONTINUA**](#_t9q2ulsihdx4) **4**

[**CÓMO INSTALAR**](#_983qchxb5fby) **5**

# **QUÉ ES JENKINS**

Jenkins es un servidor open source para la integración continua. Es una herramienta que se utiliza para compilar y probar proyectos de software de forma continua, lo que facilita a los desarrolladores integrar cambios en un proyecto y entregar nuevas versiones a los usuarios. Escrito en Java, es multiplataforma y accesible mediante interfaz web. Es el software más utilizado en la actualidad para este propósito.

# **¿QUÉ SE PUEDE HACER CON JENKINS?**

Con Jenkins podemos automatizar multitud de tareas que nos ayudarán a reducir el time to market de nuestros productos digitales o de nuevas versiones de ellos. Concretamente, con esta herramienta podemos:

* Automatizar la compilación y testeo de software.
* Notificar a los equipos correspondientes la detección de errores.
* Desplegar los cambios en el código que hayan sido validados.
* Hacer un seguimiento de la calidad del código y de la cobertura de las pruebas.
* Generar la documentación de un proyecto.

# **¿CÓMO FUNCIONA JENKINS?**

Para entender cómo funciona Jenkins vamos a ver un ejemplo de cómo sería el flujo de integración continua utilizando esta herramienta:

* Un desarrollador hace un commit de código en el repositorio del código fuente.
* El servidor de Jenkins hace comprobaciones periódicas para detectar cambios en el repositorio.
* Poco después del commit, Jenkins detecta los cambios que se han producido en el código fuente. Compila el código y prepara un build. Si el build falla, envía una notificación al equipo en cuestión. Si resulta exitoso, lo despliega en el servidor de testeo.
* Después de la prueba, Jenkins genera un feedback y notifica al equipo el build y los resultados del testeo.
* Jenkins continúa revisando el repositorio frecuentemente y todo el proceso se repite.

# **VENTAJAS DE USAR JENKINS**

Algunas de las características que hacen de Jenkins una de las mejores herramientas para aplicar integración continua son:

* Es sencilla de instalar.
* Es una herramienta opensource respaldada por una gran comunidad.
* Es gratuita.
* Es muy versátil, gracias a sus centenares de plugins.
* Está desarrollada en Java, por lo que funciona en las principales plataformas.

# **QUÉ ES LA INTEGRACIÓN CONTINUA**

La integración continua o Continuous Integration (CI) es una práctica habitual en desarrollo de software que consiste en integrar frecuentemente mejoras en el código de un proyecto una vez han sido validadas, normalmente varias veces al día, con el objetivo de detectar errores lo antes posible.

Cada cambio que realiza un desarrollador, se comprueba compilando el código fuente y obteniendo un ejecutable (llamado build). Si es validado, será incorporado al código fuente y desplegado.

Realizar builds frecuentemente y comprobar su funcionamiento ayuda a conseguir un producto final más fiable

# **CÓMO INSTALAR**

El primer paso sería irnos a la web de dockerhub, y buscamos la imagen de Jenkins de bitnami.

Una vez hecho esto, nos vamos a la parte de docker-compose y escribimos en nuestra terminal el comando que nos aparece:

* $ curl -sSL https://raw.githubusercontent.com/bitnami/containers/main/bitnami/jenkins/docker-compose.yml > docker-compose.yml

Una vez escrito este comando, escribirnos en comando con el cual lo iniciaremos:

* $ docker-compose up -d

Buscaremos la siguiente direccion: <http://localhost> , y entraremos con el username “user”, y password “bitnami”

Esta pantalla debería aparecer al entrar en la dirección anterior

