

# **HERRAMIENTAS PARA EL PROCESO DE DESARROLLO E INTEGRACIÓN CONTINUA.**

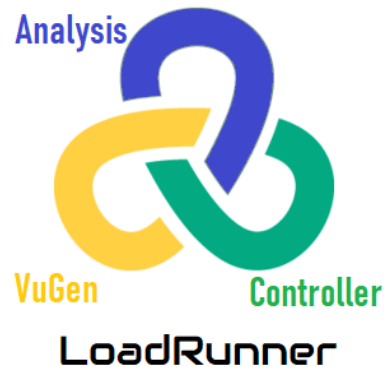
# Agenda:

- Proceso de automatización de los elementos de la Integración Continua (CI pipeline):
  - proceso de compilación (Build).
  - proceso de pruebas.
  - comunicación de resultados de las pruebas.
- Herramientas para la automatización de la Integración Continua aplicables al tipo de desarrollo:
  - Herramientas de automatización de la compilación (build).
  - Herramientas de automatización de pruebas



# Herramientas comunes de CI/CD

- Son útiles para que los equipos automaticen el desarrollo, la implementación y las pruebas.
- Hay algunas que gestionan específicamente la integración (CI), otras que abordan el desarrollo y la implementación (CD), y algunas que se especializan en las pruebas permanentes u otras funciones relacionadas.



# Herramientas para pruebas

## Objetivo:

Elegir la herramienta correcta es esencial para garantizar una implementación efectiva de las pruebas y, finalmente, para **asegurar la calidad y fiabilidad del software entregado.**





**Prueba de aceptación**

**Pruebas de integración**

**Pruebas de unidad**

**Pruebas funcionales**

**Pruebas de rendimiento**

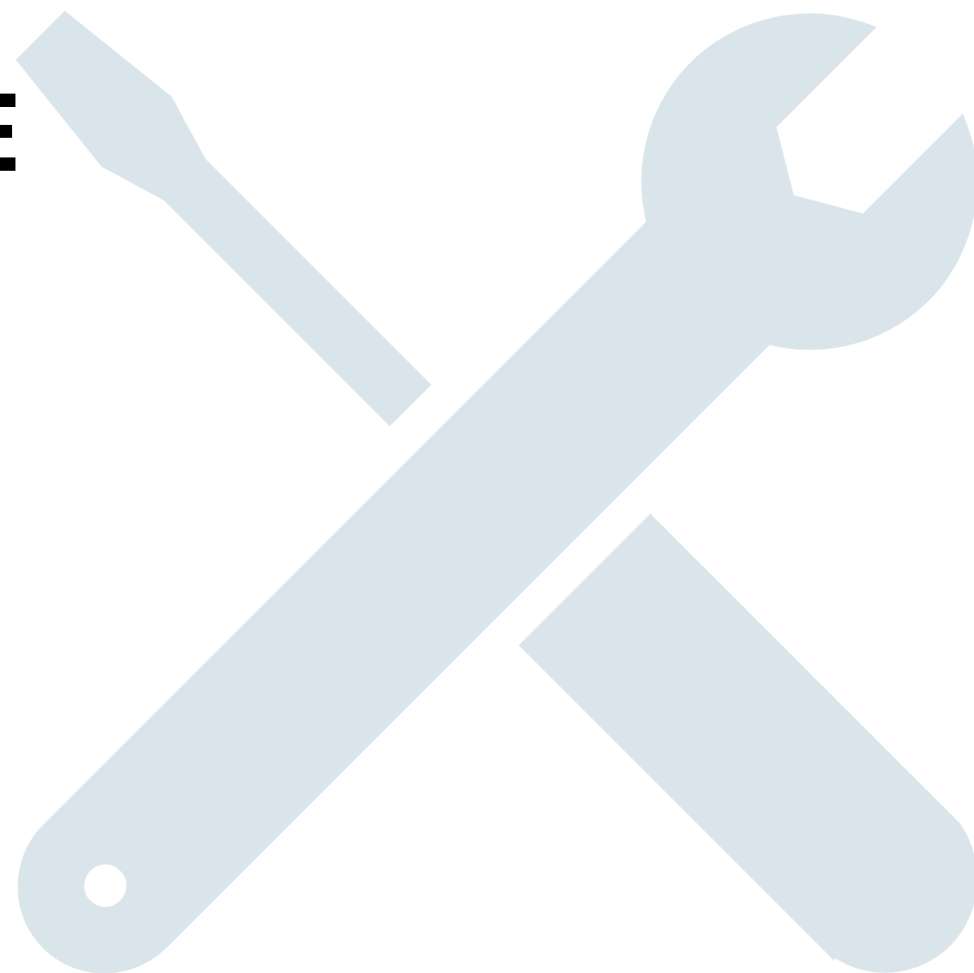
**Pruebas de regresión**

**Pruebas de estrés**

**Pruebas de usabilidad**



# **HERRAMIENTAS DE AUTOMATIZACIÓN DE LA COMPILACIÓN**





Las compilaciones automatizadas desempeñan un papel esencial en [la integración continua](#) (CI) y son un elemento clave de un [proceso de CI/CD](#).



Constituyen el primero de una serie de pasos automatizados diseñados para **alertarle lo antes posible de cualquier problema en su código causado por sus últimos cambios**.



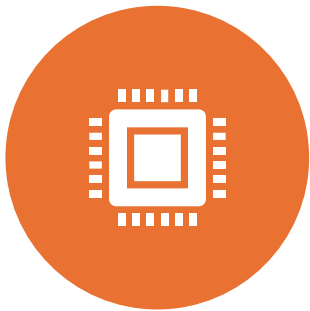
El proceso automatizado de compilación incluye una serie de comprobaciones, así como la recopilación de todas las piezas necesarias para que su programa se ejecute; incluso si está trabajando en un lenguaje interpretado, necesitará un paso de compilación.



Los archivos resultantes de la etapa de compilación -conocidos como **artefactos de compilación**- avanzan a través de su proceso de CI/CD hacia otras etapas de prueba seguidas de la puesta en escena. Si una compilación supera todos los pasos del proceso, está listo para su publicación.

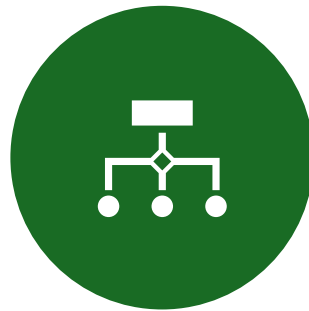


# HERRAMIENTAS DE AUTOMATIZACIÓN DE LA COMPILACIÓN:



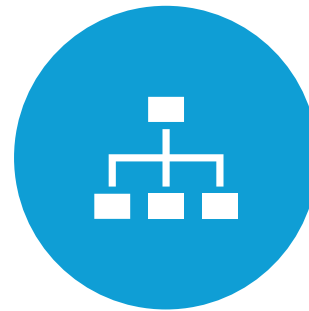
## **CAKE (C# MAKE):**

UNA HERRAMIENTA DE CÓDIGO ABIERTO QUE UTILIZA C# Y UN LENGUAJE ESPECÍFICO DE DOMINIO, COMPATIBLE CON WINDOWS, MACOS Y LINUX<sup>1</sup>.



## **MSBUILD:**

LA PLATAFORMA DE COMPILACIÓN PARA .NET Y VISUAL STUDIO<sup>1</sup>.



## **NUKE:**

PARA PROYECTOS .NET, .NET CORE Y MONO<sup>1</sup>.



## **FAKE (F# MAKE):**

UNA HERRAMIENTA DE AUTOMATIZACIÓN DE COMPILACIÓN

# HERRAMIENTAS PARA AGILIZAR LA AUTOMATIZACIÓN

más usadas



Jenkins es una herramienta utilizada en el ámbito de la **integración continua (CI)**. Utilizando Jenkins, se puede automatizar tareas como la **compilación, pruebas y despliegue** de aplicaciones de manera continua. Su flexibilidad y la gran cantidad de complementos disponibles hacen que sea una herramienta altamente personalizable y adecuada para diversos proyectos.



El **control de versiones** es fundamental en DevOps, **Git** facilita el seguimiento de cambios, la colaboración entre equipos y la integración con otras herramientas de automatización. Esta herramienta permite la **creación de ramas** para experimentar nuevas funcionalidades sin comprometer la rama principal, lo que brinda una mayor agilidad y seguridad al equipo de desarrollo.



**Docker** es una plataforma de contenedores que ha revolucionado la forma en que se desarrolla y despliega el software. Permite **empaquetar aplicaciones y sus dependencias** en entornos aislados, lo que garantiza la portabilidad y la consistencia en diferentes etapas del ciclo de vida de una aplicación. Docker facilita la **creación de entornos reproducibles** y escalables, lo que agiliza los procesos de pruebas y despliegue.



## kubernetes

A medida que las aplicaciones basadas en contenedores se vuelven más complejas, se necesita una solución para gestionar eficientemente el despliegue y escalamiento. Aquí es donde entra **Kubernetes**, una herramienta de **orquestación de contenedores** que automatiza tareas como el despliegue, la supervisión y la escalabilidad de aplicaciones en contenedores. Con Kubernetes, se puede garantizar una gestión eficiente de los recursos.



ANSIBLE

**Ansible** es una herramienta de automatización que permite **gestionar la configuración y la infraestructura** de servidores, la implementación de aplicaciones y la gestión de la infraestructura. Su enfoque es declarativo y su capacidad para realizar cambios de manera repetible y controlada hacen que Ansible sea una herramienta muy valiosa en DevOps.



## puppet

**Puppet** se utiliza para **gestionar la configuración y la infraestructura** como código. Permite automatizar la configuración de servidores y aplicaciones para garantizar la escalabilidad en el entorno de producción. Puppet utiliza un **enfoque declarativo**, lo que significa que describe el estado del sistema y se encarga de mantener esa configuración de manera automatizada.



**JFrog Artifactory** es una herramienta de **gestión de artefactos** que permite almacenar, organizar y distribuir los diferentes componentes y paquetes de software utilizados en un proyecto. Ofrece un **repositorio centralizado** y seguro para los artefactos, lo que facilita su acceso y control.



**Chef** es una herramienta que permite la configuración, el aprovisionamiento y la gestión de la infraestructura y las aplicaciones de forma consistente y escalable. Se utiliza para **implementar y mantener la infraestructura** de TI y automatiza la configuración de servidores, aplicaciones y entornos de desarrollo.



**SonarQube** es una herramienta de **análisis estático de código** que ayuda a garantizar la calidad del código en el proceso de desarrollo. Esta herramienta, además, permite **identificar y corregir problemas de código**, como vulnerabilidades, errores y duplicación, y proporciona métricas y estadísticas detalladas sobre la calidad del código. SonarQube facilita la detección temprana de errores y promueve buenas prácticas de programación, lo que facilita la entrega de aplicaciones más seguras y de mejor calidad.

# Conclusión

- Cada una de estas herramientas de automatización para DevOps desempeña un papel fundamental en el proceso de desarrollo de software, desde la integración continua hasta el despliegue y la gestión de infraestructuras.
- Al utilizar estas herramientas, los equipos de desarrollo pueden aumentar la eficiencia, mejorar la calidad del software y acelerar la entrega de aplicaciones en un entorno ágil y colaborativo.