DOCUMENTO ESPECIFICACIÓN TÉCNICA JENKINS– PROYECTO MIGRACIÓN SICOF

Fábrica de software – ADA S.A.S.

**REVISIÓN Y CONTROL DE CAMBIOS**

**Revisión y Versiones**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Autor | Versión | | Descripción | Cargo | Fecha | |
| Jhonn Edison Gomez Melo | 1.0 | Documento | | Desarrollador | | 16/08/2024 |

**Control de Cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Autor | Versión | Descripción del cambio |
| 16/08/2024 | Jhonn Edison Gomez Melo | 1.0 | Liberación Documento |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

TABLA DE CONTENIDO

Contenido

[**RESUMEN** 4](#_Toc174689429)

[**INTRODUCCIÓN** 5](#_Toc174689430)

[**INTRODUCCIÓN A LAS URLS O VARIABLES** 5](#_Toc174689431)

[**JENKINS** 6](#_Toc174689432)

[**¿QUE ES JENKINS?** 6](#_Toc174689433)

[**CARACTERÍSTICAS DE JENKINS** 6](#_Toc174689434)

[**¿QUÉ ES LA INTEGRACIÓN CONTINUA (CI)?** 7](#_Toc174689435)

[**¿POR QUÉ USAR JENKINS PARA REALIZAR CI?** 8](#_Toc174689436)

[**CONFIGURACIÓN INICIAL DE JENKINS** 9](#_Toc174689437)

[**CONCLUSIÓN** 13](#_Toc174689438)

# **RESUMEN**

Este documento proporciona una visión general de Jenkins, una herramienta de automatización popular utilizada para la integración continua (CI) en proyectos de software. Comienza explicando qué es Jenkins y sus principales características, destacando su papel crucial en los entornos de desarrollo modernos.

Luego, se define el concepto de integración continua (CI) y se explora por qué Jenkins es una opción preferida para implementar CI en proyectos. Finalmente, se incluyen instrucciones detalladas para instalar Jenkins utilizando una imagen Dockerizada, facilitando su despliegue y configuración en diferentes entornos.

# **INTRODUCCIÓN**

Jenkins es una herramienta de automatización que ha revolucionado la forma en que los equipos de desarrollo gestionan la integración continua (CI) y la entrega continua (CD).

Este documento está diseñado para proporcionar una comprensión integral de Jenkins, comenzando con una explicación de qué es Jenkins y sus características principales. Además, se explorará el concepto de integración continua (CI), destacando su importancia en el ciclo de vida del desarrollo de software. Dado que Jenkins es una herramienta ampliamente utilizada para CI, se detallarán las razones por las cuales es una opción ideal para este propósito.

Finalmente, se ofrecerá una guía paso a paso para instalar Jenkins mediante una imagen Dockerizada, permitiendo a los usuarios desplegar y configurar Jenkins de manera rápida y eficiente en sus entornos de desarrollo.

## **INTRODUCCIÓN A LAS URLS O VARIABLES**

En este documento, todas las URLs o variables utilizadas están almacenadas en la Wiki, la se encuentra en la siguiente ruta:

**Ruta de variables de Configuración:**

[**URL\_WIKI**/doku.php?id=ada:howto:sicoferp:factory:new-migracion-sicoferp:rutas\_de\_conexiones](http://10.1.20.89/doku.php?id=ada:howto:sicoferp:factory:new-migracion-sicoferp:rutas_de_conexiones)

A continuación, se listan las variables utilizadas en este documento para referirse a las URLs o variables, cabe resaltar que estas pueden cambiar dependiendo del ambiente en el que se está utilizando el enfoque:

* **URL\_JENKINS:** URL para el servidor de Jenkins.
* **USER\_JENKINS:** Usuario para el servidor de Jenkins.
* **PASS\_JENKINS:** Contraseña para el servidor de Jenkins.
* **URL\_WIKI:** URL del repositorio Wiki

# **JENKINS**



## **¿QUE ES JENKINS?**

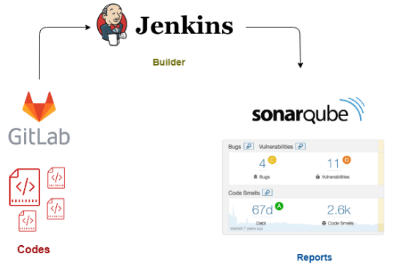
Jenkins es una plataforma de código abierto ampliamente utilizada para la Implementación Continua (CI) y la Entrega Continua (CD) en el desarrollo de software. Como una herramienta fundamental en el proceso de desarrollo moderno, Jenkins automatiza tareas repetitivas y tediosas, permitiendo a los equipos de desarrollo concentrarse en la creación de software de alta calidad.

En su núcleo, Jenkins proporciona un entorno flexible y extensible para la automatización de procesos de construcción, pruebas y despliegue. Su arquitectura modular permite a los usuarios adaptar la plataforma a sus necesidades específicas mediante la instalación de una amplia gama de complementos (plugins) disponibles en su vasto ecosistema.

## **CARACTERÍSTICAS DE JENKINS**

* **Automatización de procesos:** Jenkins permite la automatización de tareas repetitivas y manuales, como la construcción de código, la ejecución de pruebas unitarias y la implementación de aplicaciones.
* **Integración con herramientas y sistemas:** Jenkins se integra con una variedad de herramientas y sistemas utilizados en el desarrollo de software, incluyendo sistemas de control de versiones (Git, SVN), herramientas de gestión de proyectos (Jira, Trello) y plataformas de gestión de contenedores (Docker, Kubernetes).
* **Gestión de flujos de trabajo:** Jenkins facilita la creación y gestión de flujos de trabajo complejos mediante la definición de pipelines, que permiten orquestar y automatizar el proceso de desarrollo desde la integración del código hasta su despliegue en producción.
* **Escalabilidad y distribución:** Jenkins es altamente escalable y admite la distribución de la carga de trabajo a través de una red de agentes distribuidos, lo que permite ejecutar procesos de CI/CD en paralelo para mejorar la eficiencia y el rendimiento.
* **Personalización y extensibilidad:** Jenkins ofrece una amplia gama de plugins que permiten a los usuarios personalizar y extender su funcionalidad según sus necesidades específicas, desde integraciones con herramientas de terceros hasta la creación de nuevas características y funcionalidades.

## **¿QUÉ ES LA INTEGRACIÓN CONTINUA (CI)?**



La integración continua (CI) con Jenkins es un enfoque en el desarrollo de software donde los cambios en el código se integran automáticamente en un repositorio compartido varias veces al día. Jenkins es una herramienta popular de automatización de CI que facilita este proceso.

En términos prácticos, cuando un desarrollador realiza cambios en el código y los sube al repositorio, Jenkins automáticamente toma esos cambios, compila el código y ejecuta pruebas automatizadas. Si todo pasa satisfactoriamente, Jenkins puede incluso implementar automáticamente los cambios en un entorno de prueba o producción.

Esto permite a los equipos de desarrollo detectar problemas de integración y errores de código de manera temprana en el ciclo de desarrollo, lo que ayuda a garantizar la calidad del software y a acelerar el tiempo de entrega. La integración continua con Jenkins ayuda a automatizar tareas repetitivas, reduce el riesgo de errores humanos y fomenta una cultura de desarrollo ágil y colaborativo.

## **¿POR QUÉ USAR JENKINS PARA REALIZAR CI?**

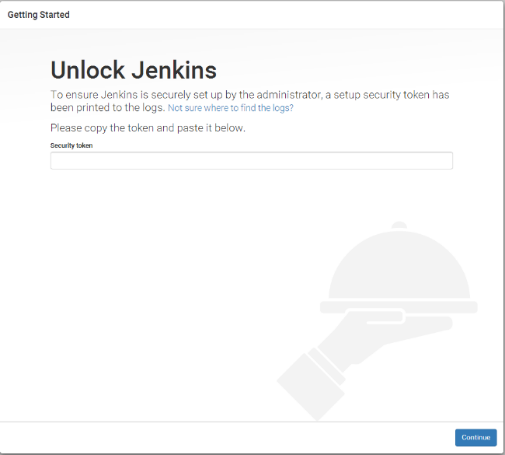
Hay varias razones por las que Jenkins es una opción popular para implementar la integración continua (CI), como, por ejemplo:

* **Flexibilidad y Personalización:** Jenkins es altamente configurable y adaptable a una amplia variedad de casos de uso y tecnologías. Ofrece una gran cantidad de complementos (plugins) que permiten extender su funcionalidad para integrarlo con herramientas y sistemas externos.
* **Código Abierto y Comunidad Activa:** Jenkins es un proyecto de código abierto con una comunidad activa de desarrolladores y usuarios que contribuyen a su mejora continua. Esto significa que hay una amplia base de conocimientos disponible y una comunidad que puede ayudar con problemas y preguntas.
* **Amplia Integración con Herramientas y Tecnologías:** Jenkins se integra fácilmente con una variedad de herramientas de desarrollo, control de versiones, sistemas de gestión de proyectos, plataformas de nube, servicios de notificación, entre otros. Esto facilita la construcción de un pipeline de CI/CD completo que se adapte a las necesidades específicas de cada equipo.
* **Escalabilidad:** Jenkins puede adaptarse a proyectos de diferentes tamaños, desde pequeños proyectos de desarrollo de software hasta grandes empresas con múltiples equipos y complejas arquitecturas de software. Su capacidad para distribuir la carga de trabajo en múltiples agentes y servidores lo hace escalable y adecuado para entornos de desarrollo de gran tamaño.
* **Historial y Confianza:** Jenkins ha estado en el mercado durante muchos años y ha sido utilizado por una amplia gama de organizaciones en todo el mundo. Su historial y la confianza que ha ganado en la comunidad de desarrollo lo convierten en una opción confiable para implementar CI.

# **CONFIGURACIÓN INICIAL DE JENKINS**

Para acceder a la interfaz de usuario de Jenkins después de la instalación del contenedor(Ver Especificación Técnica Docker) hay que acceder a la url del servidor donde se está ejecutando el contenedor y referenciar el Puerto en el cual hicimos la instalación, para este caso el Puerto es el 8083, así que nos dirigimos a un navegador y accedemos a la siguiente url: **URL\_JENKINS**

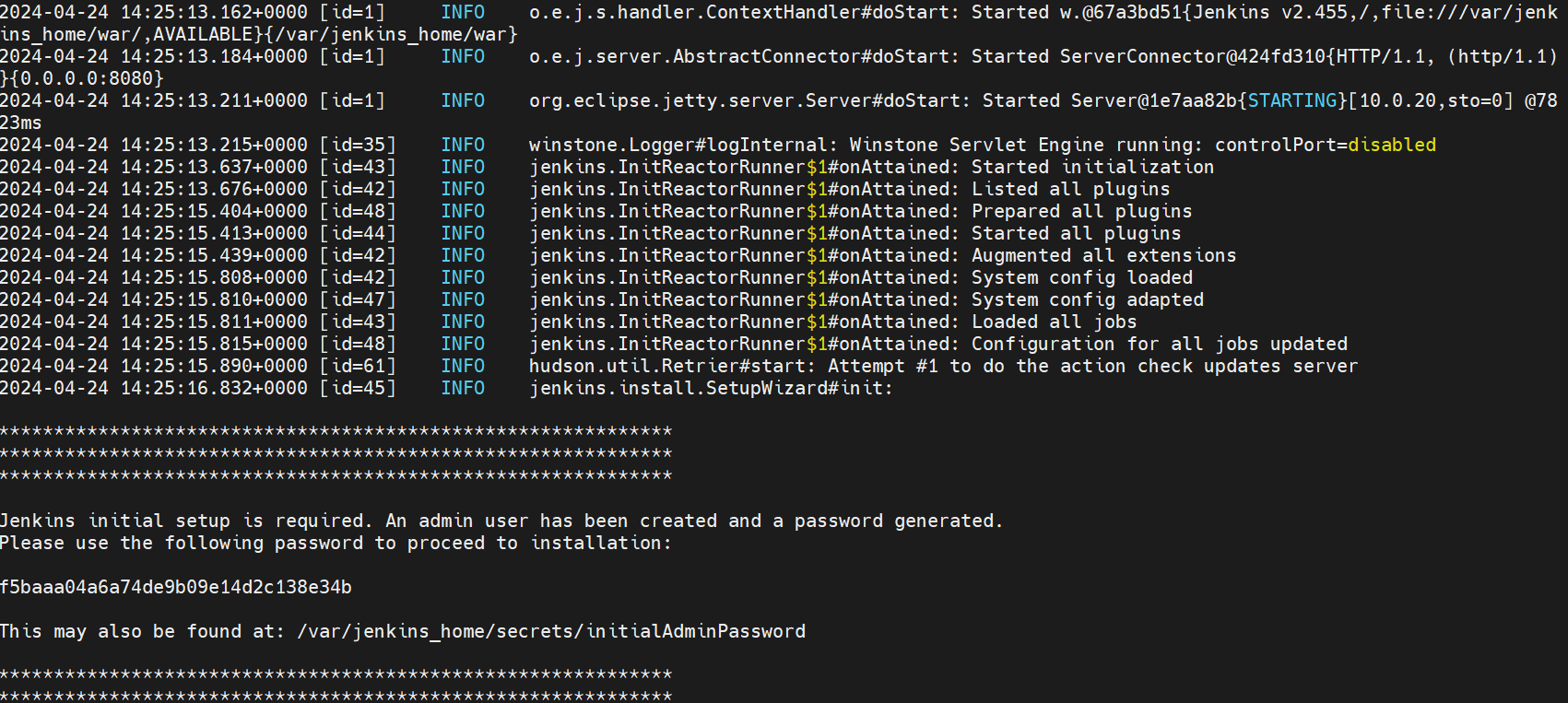
Al abrir la url Podemos ver que Jenkins se encuentra bloqueado y necesita un código de desbloqueo



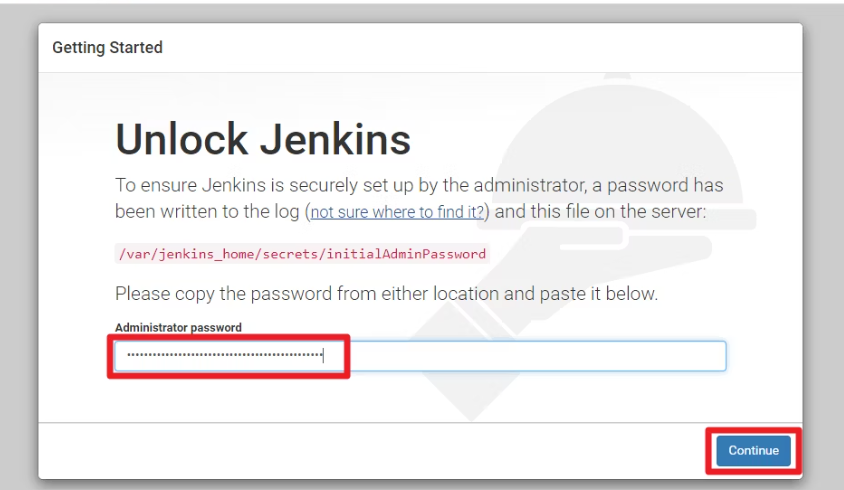
Volvemos a la consola del servidor y ejecutamos el siguiente comando:



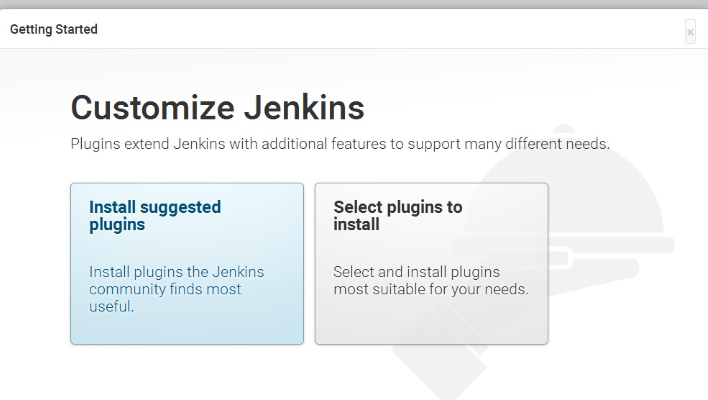
Una vez ejecutado vemos la clave que se debe ingresar para poder desbloquear Jenkins, se copia y pega en la interfaz gráfica.



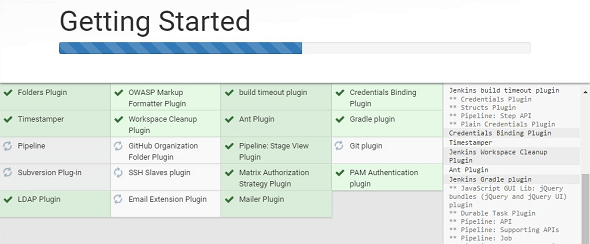
Se digita la clave que se encuentra en la consola y se da clic en el botón continuar



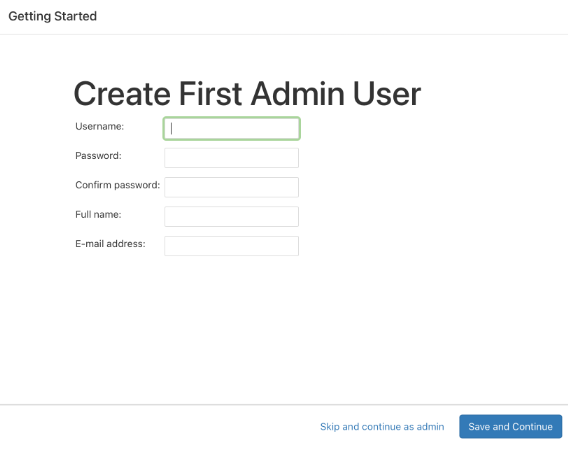
Al dar clic en continuar, Jenkins nos muestra dos tipos de instalación, una manual y la otra con los plugins sugeridos, para este caso tomaremos la instalación de plugins sugeridos



Automáticamente, Jenkins empieza a instalar los plugins sugeridos en nuestro contenedor.



Al realizar la instalación de los plugins sugeridos, Jenkins nos solicita la creación de un usuario admin para la gestión de su interfaz gráfica:



Para este caso en específico se dejan las siguientes credenciales, con las cuales se gestionará todo el entorno de jenkins:

usuario: **USER\_JENKINS**

contraseña: **PASS\_JENKINS**

En la siguiente Ventana nos muestra la url donde quedara la aplicación de Jenkins, damos clic en continuar y visualizamos la siguiente pantalla

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Esta pantalla nos indica que Jenkins fue instalado correctamente con los plugins sugeridos.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

# **CONCLUSIÓN**

Este documento ha proporcionado una visión integral de Jenkins, destacando su importancia en el ámbito de la integración continua (CI) y su relevancia en los procesos de desarrollo de software modernos. Comenzamos definiendo qué es Jenkins y describiendo sus características clave, lo que permitió entender por qué esta herramienta se ha convertido en un pilar fundamental en la automatización de pipelines de CI/CD.

A través de la explicación del concepto de integración continua, se ha demostrado cómo Jenkins facilita la detección temprana de errores, mejora la calidad del código, y acelera el ciclo de entrega de software. También se discutieron las razones por las cuales Jenkins es una opción óptima para implementar CI en proyectos de diversas escalas, respaldado por su flexibilidad, extensibilidad y la gran comunidad que lo respalda.

Finalmente, la guía de configuración inicial de Jenkins ofreció un enfoque práctico para su implementación utilizando Docker, lo que simplifica el proceso de instalación y permite a los equipos de desarrollo comenzar a aprovechar sus beneficios rápidamente.

En resumen, Jenkins no solo es una herramienta poderosa para la integración continua, sino que también es una pieza clave en la creación de un flujo de trabajo ágil y eficiente. Con la información proporcionada en este documento, los lectores están ahora mejor equipados para entender, implementar y aprovechar Jenkins en sus proyectos, dando un paso importante hacia la automatización y la optimización de sus procesos de desarrollo.