



**Automatización del proceso de desarrollo de software mediante una herramienta de integración continua (Jenkins) para el control de versionamiento en la Dirección de Tecnologías de la Información y Comunicación del Ejército.**

Tipanluisa Santo, Darío Alexander

Departamento Electrónica y Computación

Carrera de Tecnología en Computación

Monografía, previo a la obtención del título de Tecnólogo en Computación.

Ing. Mise Pasuña, José Luis

08 de septiembre del 2020



DEPARTAMENTO DE ELECTRONICA Y COMPUTACIÓN

CARRERA DE TECNOLOGIA EN COMPUTACIÓN

CERTIFICACIÓN

Certifico que el trabajo Monográfico, "AUTOMATIZACIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE MEDIANTE UNA HERRAMIENTA DE INTEGRACIÓN CONTINUA (JENKINS) PARA EL CONTROL DE VERSIONAMIENTO EN LA DIRECCIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN DEL EJÉRCITO." fue realizado por el señor TIPANLUISA SANTO, DARÍO ALEXANDER, el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 11 de septiembre de 2020

Ing. Míse Pasaña, José Luisa

C.C.: 0502965247

## REPORTE DE VERIFICACIÓN

**URKUND**

**Document Information**

Analyzed document Trabajo de titulación Sr. Tupaluisa Santo Dario Alexander.docx (D78632135)  
 Submitted 9/4/2020 11:39:00 PM  
 Submitted by  
 Submitter email jrcaiza@espe.edu.ec  
 Similarity 1%  
 Analysis address jrcaiza.espe@analysis.arkund.com

**Sources included in the report**

<b>W</b>	URL: <a href="https://www.mti.cl/wp-content/uploads/2018/12/Tesina_2018_Cortez-Rodrigo.pdf">https://www.mti.cl/wp-content/uploads/2018/12/Tesina_2018_Cortez-Rodrigo.pdf</a> Fetched: 10/22/2019 8:46:24 AM	 1
<b>SA</b>	<b>3798.Núñez Ramírez, Hugo Sergio_.pdf</b> Document 3798.Núñez Ramírez, Hugo Sergio_.pdf (D51086249)	 1
<b>SA</b>	<b>Tercer avance.docx</b> Document Tercer avance.docx (D61374741)	 1
<b>SA</b>	<b>OCAMPO QUISINTUÑA DIEGO.pdf</b> Document OCAMPO QUISINTUÑA DIEGO.pdf (D41590934)	 2

  
 Sosa Luis Hise Pasuña  
 C.C. 0302963297



DEPARTAMENTO DE ELECTRONICA Y COMPUTACION

CARRERA DE TECNOLOGIA EN COMPUTACION

AUTORÍA DE RESPONSABILIDAD

Yo, TIPANLUISA SANTO, DARÍO ALEXANDER declaro que el contenido, ideas y criterios del trabajo de titulación: "AUTOMATIZACIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE MEDIANTE UNA HERRAMIENTA DE INTEGRACIÓN CONTINUA (JENKINS) PARA EL CONTROL DE VERSIONAMIENTO EN LA DIRECCIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN DEL EJÉRCITO." de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Latacunga, 11 de septiembre de 2020

Tipanluisa Santo, Darío Alexander

C.C.: 0503426017



DEPARTAMENTO DE ELECTRONICA Y COMPUTACIÓN

CARRERA DE TECNOLOGIA EN COMPUTACIÓN

**AUTORIZACIÓN**

Yo, TIPANLUISA SANTO, DARIO ALEXANDER, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: "AUTOMATIZACIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE MEDIANTE UNA HERRAMIENTA DE INTEGRACIÓN CONTINUA (JENKINS) PARA EL CONTROL DE VERSIONAMIENTO EN LA DIRECCIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN DEL EJÉRCITO." en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 11 de septiembre de 2020

Tipanluisa Santo, Darío Alexander

C.C. 0503426017

## **DEDICATORIA**

El presente trabajo de titulación lo dedico a Dios todo poderoso por brindarme la salud y vida para seguir en mi carrera de las armas, mi esposa e hija quienes son el pilar y soporte fundamenta de mi vida profesional, por su compañía y motivación en este capítulo de mi vida. A mis padres por el apoyo incondicional.

**Darío Alexander Tipanluisa Santo**

## **AGRADECIMIENTO**

Enormemente a Dios por sus bendiciones y guiarme en esta noble profesión de las armas, a mis queridos padres que me dieron la vida, a mi Esposa Jessica Yanchaguano que ha sido mi apoyo fundamental en este proceso y ser parte de mi vida.

**Darío Alexander Tipanluisa Santo**

## ÍNDICE DE CONTENIDO

<b>CARATULA</b> .....	<b>1</b>
<b>CERTIFICACIÓN</b> .....	<b>2</b>
<b>REPORTE DE VERIFICACIÓN</b> .....	<b>3</b>
<b>AUTORIA DE RESPONSABILIDAD</b> .....	<b>4</b>
<b>AUTORIZACIÓN</b> .....	<b>5</b>
<b>DEDICATORIA</b> .....	<b>6</b>
<b>AGRADECIMIENTO</b> .....	<b>7</b>
<b>ÍNDICE DE CONTENIDO</b> .....	<b>8</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>10</b>
<b>RESUMEN</b> .....	<b>11</b>
<b>ABSTRAC</b> .....	<b>12</b>
<b>CAPÍTULO I</b> .....	<b>13</b>
<b>1. DEFINICIÓN DEL PROBLEMA</b> .....	<b>13</b>
1.1 Tema.....	13
1.2 Antecedentes .....	13
1.3 Planteamiento del Problema .....	15
1.4 Justificación.....	16
1.5 Objetivo General .....	17
1.6 Objetivos Específicos.....	17
1.7 Alcance.....	18
<b>CAPÍTULO II</b> .....	<b>20</b>
<b>2. MARCO TEÓRICO</b> .....	<b>20</b>
2.1 Automatización .....	20
2.2 Jenkins.....	20
2.3 Control de Versiones.....	21
2.4 Especificación de las versiones .....	21
2.4.1 Versiones X.Y.Z .....	22
2.5 Tecnologías de la información y comunicación .....	22
2.6 Integración Continua .....	23
2.7 Comunicaciones del Ejercito.....	25
2.8 Java.....	26

2.9 Entornos de Desarrollo .....	27
2.9.1 NetBeans.....	27
2.10 Metodologías Agiles.....	28
2.11 DevOps.....	28
2.12 Fases de DevOps .....	29
<b>CAPÍTULO III .....</b>	<b>31</b>
<b>3. INVESTIGACIÓN E IMPLEMENTACIÓN .....</b>	<b>31</b>
3.1. Introducción.....	31
3.2. Recomendaciones de hardware para instalación de Jenkins .....	31
3.3. Requerimientos básicos para instalación de Jenkins.....	32
3.4. Instalar y configuración Jenkins en Windows server 2016 .....	32
3.7. Instalación de subversión con tortoise svn.....	38
3.8. Creación de repositorio.....	40
3.9. Creación de usuario y subida de proyectos al repositorio.....	41
3.10. Integración de Jenkins y subversión .....	44
<b>CAPÍTULO IV .....</b>	<b>48</b>
<b>4. FUNCIONAMIENTO.....</b>	<b>48</b>
4.1. Verificación del funcionamiento del servidor de automatización Jenkins .....	48
4.2. Control del historial de tareas.....	49
<b>CAPÍTULO V .....</b>	<b>51</b>
<b>5. CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>51</b>
5.1 Conclusiones .....	51
5.2 Recomendaciones.....	52
<b>REFERENCIAS BIBLIOGRAFICAS .....</b>	<b>53</b>

## ÍNDICE DE FIGURAS

<b>Figura 1</b> <i>Integración continua con Subversión</i> .....	<b>24</b>
<b>Figura 2</b> <i>Aplicando la metodología DevOps</i> .....	<b>29</b>
<b>Figura 3</b> <i>Página oficial para la descarga Jenkins</i> .....	<b>33</b>
<b>Figura 4</b> <i>Selección de sistema Operativo</i> .....	<b>34</b>
<b>Figura 5</b> <i>Instalación de Jenkins</i> .....	<b>35</b>
<b>Figura 6</b> <i>Contraseña para la instalación</i> .....	<b>36</b>
<b>Figura 7</b> <i>Instalación de Jenkins</i> .....	<b>37</b>
<b>Figura 8</b> <i>Creación de usuario y contraseña</i> .....	<b>37</b>
<b>Figura 9</b> <i>Finaliza la instalación Jenkins Windows server</i> .....	<b>38</b>
<b>Figura 10</b> <i>Instalación de TortoiseSVN</i> .....	<b>39</b>
<b>Figura 11</b> <i>Interfaz gráfica de SVN</i> .....	<b>39</b>
<b>Figura 12</b> <i>Creación de repositorios</i> .....	<b>40</b>
<b>Figura 13</b> <i>Dirección del repositorio</i> .....	<b>41</b>
<b>Figura 14</b> <i>Creación de usuario</i> .....	<b>42</b>
<b>Figura 15</b> <i>Dirección de repositorio</i> .....	<b>43</b>
<b>Figura 16</b> <i>Subida de proyecto</i> .....	<b>43</b>
<b>Figura 17</b> <i>Creación de proyecto</i> .....	<b>44</b>
<b>Figura 18</b> <i>Configuración de Jenkins con subversión</i> .....	<b>45</b>
<b>Figura 19</b> <i>Configuración de disparos de ejecución</i> .....	<b>46</b>
<b>Figura 20</b> <i>Construcción del proyecto</i> .....	<b>47</b>
<b>Figura 21</b> <i>Verificación de ejecución</i> .....	<b>47</b>
<b>Figura 22</b> <i>Ingreso al servidor</i> .....	<b>48</b>
<b>Figura 23</b> <i>Consultas al repositorio</i> .....	<b>49</b>
<b>Figura 24</b> <i>Ejecución</i> .....	<b>50</b>

## RESUMEN

En el presente trabajo investigativo, presenta la Automatización del proceso de desarrollo de software mediante una herramienta de integración continua (Jenkins) para el control de versionamiento en la Dirección de Tecnologías de la Información y Comunicación del Ejército, con la ayuda del repositorio subversión que trabajan con este servidor de automatización se podrá lograr realizar la integración continua de los sistemas que existen en Dtics del ejército ecuatoriano. Jenkins es una herramienta de construcción e integración continua que requiere que los desarrolladores compartan código mediante un repositorio, el sistema de control de versiones es monitoreado cuando un cambio es detectado automáticamente. Permitiendo que los equipos detectar errores tempranamente y así poder solucionarlo de mejor manera de tal forma que la integración continua es muy importante permite la detección de problemas tan pronto sea posible durante el desarrollo de proyecto y el resultado es un software de mejor calidad, ayuda a la calida de código evaluando continuamente y una vez verificado pasarlas a producción. Para poder realizar la integración la metodología la mejor metodología que se adapto es la DevOps la cual nos permite derribar los muros entre los departamentos de Desarrollo Y Operaciones fomentando la comunicación mediante reuniones periódicas, compartiendo archivos, automatizando procesos y facilitando la integración continua.

### Palabras claves:

- **INTEGRACIÓN CONTINUA**
- **REPOSITORIO**
- **SOFTWARE**
- **DEVOPS.**

## **ABSTRAC**

In the present investigative work, he presents the Automation of the software development process through a continuous integration tool (Jenkins) for versioning control in the Army's Information and Communication Technologies Directorate, with the help of the subversion repository that works With this automation server it will be possible to achieve the continuous integration of the systems that exist in Dtics of the Ecuadorian army. Jenkins is a continuous integration and construction tool that requires developers to share code through a repository, the version control system is monitored when a change is automatically detected. Allowing teams to detect errors early and thus be able to solve them in a better way in such a way that continuous integration is very important allows the detection of problems as soon as possible during project development and the result is better quality software, helps to The warm code continuously evaluating and once verified, pass them to production. To be able to carry out the integration of the methodology, the best methodology that is adapted is DevOps, which allows us to tear down the walls between the Development and Operations departments, promoting communication through periodic meetings, sharing files, automating processes and facilitating continuous integration.

Key words:

- **CONTINUOUS INTEGRATION**
- **REPOSITORY**
- **SOFTWARE**
- **DEVOPS.**

## CAPÍTULO I

### 1. DEFINICIÓN DEL PROBLEMA

#### 1.1 Tema

Automatización del proceso de desarrollo de software mediante una herramienta de integración continua (Jenkins) para el control de versionamiento en La Dirección de Tecnologías de la Información y Comunicación del Ejército.

#### 1.2 Antecedentes

En la actualidad existen herramientas básicas que han sido aplicadas y utilizadas para la mejorar la calidad y solución de problemas. Cada organización adopta y utiliza las que ayuden a mejorar su gestión, el éxito de estas herramientas radica en la capacidad que han demostrado para ser aplicadas en un amplio conjunto de problemas, pero si primero para hablar de herramientas continua debemos hablar de DEVOPS proviene de desarrollo y operaciones, que se refiere a una metodología de desarrollo de software que se centra en la comunicación, colaboración e integración entre desarrolladores de software y los profesionales de sistemas en las tecnologías de la información .

En Ejército ecuatoriano existen sistemas que agilitan el trabajo a los usuarios. Dentro de la ingeniería del software, se encuentra la integración continúa, de desarrollo de software donde los miembros de un equipo integran su trabajo con frecuencia, en general, cada persona integra su trabajo, al menos una vez al día, lo que conlleva a múltiples integraciones por día. Cada integración se verifica mediante una compilación automatizada, incluidas las pruebas, con el objetivo de detectar errores de integración de manera rápida.

Existen algunos trabajos relacionados, lo cual exponemos a continuación:

De acuerdo a Orozco García Ana María (2015) , con su trabajo de investigación cuyo tema es: “LA INTEGRACIÓN CONTINUA Y SU APORTE AL ASEGURAMIENTO DE LA CALIDAD EN EL CICLO DE VIDA DEL DESARROLLO DE SOFTWARE”, menciona que actualmente el desarrollo de aplicaciones se encuentra apoyado por diferentes normas, certificaciones de procesos, además prácticas y herramientas que aportan mejoras en el diseño, implementación y desarrollo de software, esta investigación se centra en una de las prácticas que se vienen implementando las empresas para agilizar los procesos de desarrollo y controlar la calidad del mismo.

De acuerdo Julio Alberto Fernández Guerrero (2017) con su trabajo de investigación cuyo tema es: “IMPLANTACIÓN DE UN SISTEMA DE INTEGRACIÓN CONTINUA EN UNA METODOLOGÍA CONSOLIDADA” menciona que la integración continua se basa en la construcción automática de proyectos con frecuencia alta, promoviendo la detección de errores en un momento temprano para poder dar prioridad a corregir dichos errores.

La Dirección de Tecnologías de la Información y Comunicación del Ejército está introduciendo cambios en la arquitectura de las aplicaciones y tecnologías de los nuevos proyectos de desarrollo, oportunidad que se desea aprovechar para implementar un proceso automatizado para la entrega continua de software alineado a una metodología de desarrollo con enfoque ágil.

Por lo tanto, se hace necesaria la implantación de un sistema de integración continua con la herramienta (Jenkins) dentro de La Dirección de las Tecnologías de la información Comunicación del Ejército Ecuatoriano para alcanzar el mayor grado de automatización posible en sus pruebas de proyectos permitiendo así a los desarrolladores ahorrar tiempo en estas tareas para centrarse en otras, surgiendo así la necesidad de realizar un análisis, diseño y automatización de un sistema de integración continua dentro de la DTics..

### 1.3 Planteamiento del Problema

La creación del arma de comunicaciones el 15 de noviembre 1962 el Ejército Ecuatoriano, y La Dirección de Tecnologías de Información y Comunicación Fuerza Terrestre, ha contribuido indiscutiblemente a la edificación del Ecuador democrático y soberana la misma que es encarga de los sistemas, desarrollan productos de software para sistematizar y agilizar el proceso de reclutamiento para administración del personal entre otros y existen inconvenientes en la calidad de sus productos. Si bien, algunas cuentan con herramientas que ayudan a mejorar ciertos tropiezos dentro de la cadena productiva del software, como lo son los servidores para controlar las versiones de la aplicación o las aplicación es que se encuentren desarrollando, siguen existiendo problemas al momento de ejecutar el paso a producción al tratar de desplegar una versión donde se encuentren involucrados más de un desarrollador; también pueden contar con herramientas donde se encuentren automatizadas las pruebas unitarias y de calidad, pero que al solucionar incidentes que se presentan, se pierde el control de la versión o se afectan otras partes de la aplicación, aumentando la posibilidad de encontrar fallos en los despliegues. Al estar vinculados en un mismo proyecto varios desarrolladores, los cuales tiene formas diferentes de realizar su código, seguramente se enfrentarán a reprocesos más adelante, en el momento de realizar mantenimientos adaptativos, evolutivos o perfectivos, por no contar con estándares de calidad definidos por la Dirección de Comunicaciones del Ejército, y realizar inspecciones continuas de código a través de una herramienta para detectar problemas de calidad en él:

Los problemas descritos afectan directamente los costos de un proyecto de software, al aumentar los tiempos estimados para el desarrollo del producto.

Se pierde el control en la fase del desarrollo del producto, pasando a través de los diferentes ambientes, de pruebas a producción, devolviéndose de producción a desarrollo, de desarrollo a pruebas, generando impacto negativo para el ordenamiento del proyecto.

Por lo antes mencionado es necesario la automatización del proceso de desarrollo de software mediante una herramienta de integración continua Jenkins en el uso de herramientas tecnológicas que permita agilizar el trabajo de los desarrolladores de la Dirección de Tecnologías de Información y Comunicación Fuerza Terrestre para que los sistemas sean de mejor calidad, estén presentados en menor tiempo posible, además que se pueda realizar el control de versionamiento.

#### **1.4 Justificación**

De acuerdo con la evolución del desarrollo de software en los últimos años, es fundamental que. La Dirección de Tecnologías de Información y Comunicación Fuerza Terrestre cuente con una herramienta de integración continua, (Jenkins) la cual es necesaria para los desarrolladores puedan agilizar su trabajo también:

Ayudará a compartir información, automatizar procesos, reducir el tiempo de implementación de los sistemas de la Fuerza terrestre.

Desarrollar software de alta calidad más rápido, ya que Jenkins es que es un servidor de integración continua de código abierto escrito en Java.

Reducir la complejidad de proyectos largos, reducir errores de los sistemas que se realizan en la Dtics.

Mejorar los costes de productividad ahorro significativo de mantenimiento y soporte.

Se beneficiarán del presente trabajo investigativo las autoridades, oficiales y voluntarios desarrolladores de La Dirección de Tecnologías de Información y Comunicación de la fuerza Terrestre Fuerza Terrestre.

Los resultados obtenidos permitirán mejora la automatización de tareas como descargar código fuente, construir y testear, los desarrolladores pueden enfocarse en realizar las nuevas funcionalidades que los usuarios demandan.

Por lo expuesto es importante que la Fuerza Terrestre particularmente de La Dirección de Tecnologías de Información y Comunicación, Automatice del proceso de desarrollo de software mediante una herramienta de integración continua (Jenkins) para el control de versionamiento, de los sistemas y así poder desarrollar software de alta calidad y reducir los errores de los sistemas.

### **1.5 Objetivo General**

Automatizar el proceso de desarrollo de software mediante una herramienta de integración continua (Jenkins) para el control de versionamiento en La Dirección de Tecnologías de la Información y Comunicación del Ejército.

### **1.6 Objetivos Específicos**

- Establecer el análisis de arquitectura de la herramienta de integración continua Jenkins para un entorno desarrollo de Software.
- Implementar una metodología para la práctica de Integración Continua con la herramienta Jenkins en un entorno desarrollo de Software en el proceso de aseguramiento de la calidad de sistemas para La Dirección de Tecnologías de Información y Comunicación de la fuerza terrestre Fuerza Terrestre.

- Realizar las respectivas pruebas de integración Continua con la herramienta Jenkins en un entorno desarrollo de software.

### **1.7 Alcance**

El presente proyecto tiene como finalidad principal la automatización del proceso de desarrollo de software mediante una herramienta de integración continua (Jenkins) para el control de versionamiento en La Dirección de Tecnologías de Información y Comunicación Fuerza Terrestre, de tal manera se agilizaría el trabajo de los desarrolladores y los sistemas sean de mejor calidad y en menor tiempo posible. La automatización del proceso de desarrollo de software mediante una herramienta de integración continua (Jenkins). Ayudará a compartir información, automatizar procesos, reducir el tiempo de implementación de los sistemas de la Fuerza Terrestre, así modelo de implementación de Integración Continua, llegando hasta diseñar una interacción que facilite el entendimiento y la aplicación de esta práctica aportando al aseguramiento de la calidad del software, al desarrollo software de alta calidad más rápido, ya que Jenkins es un servidor de integración continua de código abierto escrito en Java. Permite reducir la complejidad de proyectos largos, reducir errores de los sistemas que se realizan en la Dtics.

Al finalizar el proyecto y finalmente al Automatizar del proceso de desarrollo de software mediante una herramienta de integración continua (Jenkins) para el control de versionamiento en La Dirección de Tecnologías de la Información y Comunicación los desarrolladores podrán revisar las métricas de calidad del software establecidas por el equipo de trabajo, enviar las modificaciones del software, una vez pasadas todas las validaciones, al repositorio principal, automatizar la compilación del software o su despliegue, una vez se hayan integrado nuevos cambios en el proyecto, notificar debidamente a los desarrolladores o al equipo de aseguramiento

de la calidad cuando se encuentra cualquier tipo de error, ya sea en base a las pruebas del software o a las métricas de calidad definidas, y así facilitar el trabajo a los desarrolladores de La Dirección de Tecnologías de la Información Comunicación de Ejército Ecuatoriano.

## CAPÍTULO II

### 2. MARCO TEÓRICO

Este trabajo investigativo va a definir la Automatización del proceso de desarrollo de software mediante una herramienta de integración continua (Jenkins) para el control de versionamiento en La Dirección de Tecnologías de la Información y Comunicación del Ejército

#### 2.1 Automatización

Se puede definir a la automatización como un conjunto de elementos, equipos y también sistemas de información y de procedimiento interrelacionales funcionando entre sí que componen una estructura jerárquica propagada cuya función es probar el funcionamiento a través de la operación de control y a la vez inspeccionando el total del sistema (Izaguirre, 2012)

Forma de ordenación a partir de la secuenciación automática de tareas y regulación de variables para que sigan las consignas impuesta (Álvarez, 2005).

En si podemos decir que la automatización es de suma importancia en la actualidad para poder agilizar de mejor manera los procesos y así poder ahorrar costos y tiempo de los sistemas que se realizan los desarrolladores.

#### 2.2 Jenkins

Es una aplicación que ayuda a los desarrolladores a realizar integración continua de proyectos de software y ayuda detectar y corregir errores de integración de forma continua. Fue creado del desarrollo original Hudson. Esta herramienta está implementada en lenguaje de programación Java por lo siguiente para poder utilizarlo es necesario una máquina virtual (Avantaggiato, Castro, & Loiacono, 2013).

Según (PABLO, USAMENTIAGA FERNANDEZ , & BARREGO , 2019) dice que se trata de uno de los servidores de integración continua más conocida y extendidas, del proyecto empezó con Hudson, pero termino siendo abandonado y se empezó a utilizar un fork llamado Jenkins.

En si Jenkins es una herramienta de integración continua que nos ayuda a localizar a corregir y detectar errores de forma rápida y continua y que es útil para los diferentes lenguajes de programación y ayuda a facilitar a los desarrolladores para que puedan realizar su trabajo de mejor manera y de una forma rápida y disminuyendo costos y tiempo.

### **2.3 Control de Versiones**

Según (Borrell, 2006) menciona que control de versiones está considerado como un método y también una herramienta que puede ser utilizado para controlar todo lo referente a los cambios que se realizar de un registro.

Al momento que existen dificultades de un registro de código de texto terminado con la primera escritura; debemos realizar cambios o reescrituras para solucionar errores, y de forma ágil modificar su contenido. El control de versiones es un técnico estándar para mantener esta memoria haciendo además que sea útil para el desarrollo futuro.

### **2.4 Especificación de las versiones**

Para poder especificar las versiones es recomendable utilizar un método más frecuente de asignar una versión a un documento es mediante un número o un grupo de números. No existe un número preciso de numerar una versión, se deja al criterio de cada desarrollador. Sí existen ciertas prácticas habituales en la numeración como el uso de tres cifras o la numeración decimal (Borrell, 2006).

### **2.4.1 Versiones X.Y.Z**

Para realizar las versionamiento es tradicional numerar las versiones es utilizando dos o tres cifras decimales para indicar la importancia de los cambios realizados. El cambio de la primera cifra indica cambios más importantes que el de la segunda. El criterio más habitual es seguir las siguientes normas:

- La primera cifra (X) indica la versión mayor del documento. Si empieza con un cero significa que el documento aún no está listo o no cumple con los requerimientos mínimos. Cada cambio en esta cifra denota una reescritura o la incompatibilidad con versiones mayores anteriores.
- La segunda cifra (Y) indica la versión menor del documento. Denota cambios en el contenido o en la funcionalidad del documento, pero no lo suficientemente importantes como para decir que ya no es el mismo. Cuando se estrena una versión mayor se deja la versión menor a cero, pero aun así se incluye de modo que la segunda versión mayor sería la 2.0.

- La tercera cifra (Z) indica la segunda versión menor. Indica que el documento se ha corregido pero que no se ha añadido ni eliminado nada relevante. Cuando se estrena una versión menor, es decir, cuando la segunda versión menor es igual a cero; suele omitirse.

- Pueden añadirse recursivamente versiones menores, algunos proyectos hacen uso de ellas cuando los criterios de selección de versiones no corresponden con los definidos en los puntos anteriores (Borrell, 2006).

## **2.5 Tecnologías de la información y comunicación**

Las TIC (Tecnologías de la Información y Comunicaciones) son las tecnologías las más utilizadas y necesarias para la tarea y evolución de la información que se necesitan para la gestión y transformación de los datos, y en particular el uso de computadoras y softwares que permiten crear, modificar, almacenar, proteger y recuperar esa información. Sin embargo, las

computadoras son lo más esencial para la identificación, selección y registro de la información (Sánchez Duarte, 2008).

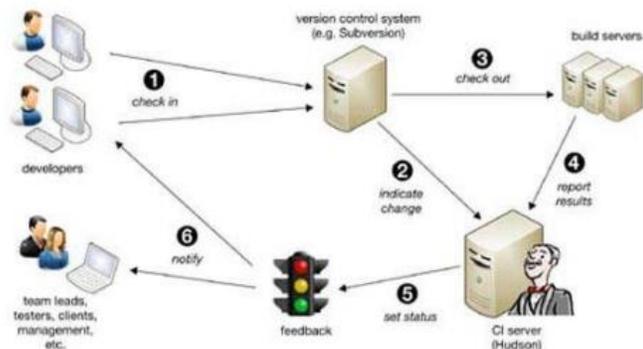
Según (ÁLVAREZ & CANTÓN MAYO , 2009) definen que las TIC son una realización social que facilitan los procesos de información y comunicación, gracias a los diversos desarrollos tecnológicos, en áreas de una construcción y extensión del conocimiento que derive en la satisfacción de las necesidades de los integrantes de una determinada organización social.

La tecnología de la información cada día avanza y está cambiando constantemente en los ámbitos sociales sin embargo eso ayuda para que el usuario gracias a su evolución se le facilite el trabajo en las comunicaciones y los sistemas informáticos.

## **2.6 Integración Continua**

Según (OROZCO, 2015) domina que la integración continua es una práctica que se realiza al desarrollo de software realizado con metodologías ágiles, que se puede elaborar siempre y cuando se implementen automatizaciones en sus procesos de requerimientos, diseño, desarrollo, pruebas y despliegues, pueden existir cambios en el código que se pueden compilar el mismo día y, las veces que sean necesarias lo que, siendo reglamentario que todos los desarrolladores del equipo integren su parte de código por lo menos una vez al día y así puedan trabajar de mejor manera.

Figura 1

*Integración continua con Subversión*

Nota: Este gráfico indica el escenario real de la integración continua y los ciclos. Tomada de (OROZCO, 2015)

Se pueden definir modelos para la implementación de integración continua para procesos de los cuales son básicamente para el proceso de codificación de software, donde se podría realizar del siguiente orden:

1.- El programador ejecuta un (cambio) sobre el SCM (Software Configuration Managemen).

2.-. Posteriormente del (cambio) el administrador de integración detecta el cambio, toma del repositorio las últimas versiones y ejecuta los scrips que forman todo el Software.

3.-. El integrador de Administración continua informa por e-mail a cerca de los resultados de la compilación a los miembros del grupo configurados en él.

4.- Finalmente el administrador realiza inspección con una periodicidad determinada. La implantación de IC, trae para una organización un cambio impactante en el área de mantenimiento y desarrollo de software, ya que introduce altos estándares de calidad en la

programación, trayendo un crecimiento en la calidad del producto y de los profesionales involucrados. (OROZCO, 2015).

(Bersabé, 2016) Afirma que Integración Continua se centra en tener un feedback casi instantáneo del estado de cada entrega de código al repositorio, a través de la automatización del proceso de construcción del software. Nos da la posibilidad de detectar fallos de integración en las primeras etapas de la construcción de nuestro software, aumentando notablemente la calidad el software entregado.

Integración continua se puede considerar como una práctica que se realiza al desarrollo de software mediante una metodología ágiles y se pueden realizar la automatización de sus procesos de igual manera podemos encontrar errores de integración en las primeras etapas de la creación de los sistemas y así poder realizar de mejor manera.

## **2.7 Comunicaciones del Ejercito**

El Arma de Comunicaciones, como arma de apoyo de combate, se torna indispensable para el ejercicio del mando, la coordinación y el control de las operaciones militares. En tiempos de guerra, es la primera en entrar y la última en salir; y en tiempos de paz, es la responsable de apoyar eficientemente la ejecución de las actividades administrativas de la institución en la conducción militar, utilizando todos sus medios.

Esta Arma tuvo un proceso de crecimiento, desarrollo y llegó a una etapa en la que, como efecto del desarrollo y el fortalecimiento del Ejército, era imperiosa la creación de una especialidad. Su entrenamiento, equipamiento y misión debían posibilitar el ejercicio del comando y el control.

El 14 de junio de 1910, el Presidente Constitucional de la República del Ecuador, General Eloy Alfaro Delgado, consideró la necesidad de reglamentar y organizar las

comunicaciones militares. Para tal efecto, emitió el primer decreto ejecutivo relacionado con esta importante actividad. Estableció la conformación de la Plana Mayor de las secciones de electricistas-telegrafistas en campaña. Del contenido de este decreto y de los tres que le precedieron, en este período apareció la primera Unidad Militar de Transmisiones al servicio del Cuartel General del Ejército. La que, con el pasar del tiempo, se constituyó en la actual Dirección de Sistemas de Información y Comunicaciones del Ejército.

Para 1929, el presidente de la República, Dr. Isidro Ayora, emitió el Decreto Ejecutivo 164, con fecha 5 de agosto. En este creó el Batallón de Ingenieros Nro. 2 "CHIMBORAZO", compuesto por una plana mayor y tres compañías, la tercera fue de Transmisiones. Con este nuevo orgánico, desapareció la unidad independiente que se encontraba adscrita al Estado Mayor General del Ejército.

Por convenio entre la República del Ecuador y el Gobierno de Estados Unidos, en el año 1943, inició sus actividades la Misión Militar Norteamericana. Este organismo fue el encargado de coordinar y ejecutar el apoyo militar norteamericano a nuestras Fuerzas Armada (Ecuatoriano, 2020)

## **2.8 Java**

Afirma (Allende, 2005) que java es un lenguaje de programación moderno que fue , mostrado por primera vez por Sun Microsystems en el segundo semestre de año 1995, Desde sus principios ganó seguidores rápidamente por muy buenas razones, y una de las más importantes es su neutralidad respecto de la plataforma de ejecución lo que permite, entre otras cosas, añadir programas a una página Web.

En si lo que más les guste a los programadores de la programación java son un de aspectos que le hacen muy satisfactorio y agradable de usar para programar son las siguientes:

La candidez y distinción de cómo se escriben los programas en Java, A ello se Line que es un lenguaje orientado a objetos que evita muchas preocupaciones a los programadores, En el proceso de compilación se realizan multitud de comprobaciones que permiten eliminar muchos posibles errores posteriores.

Las bibliotecas ya definidas que proporciona el lenguaje y que el programador puede utilizar sin tener que hacerlas de nuevo.

Según (Sierra, 2015) Java es un lenguaje de programación de alto nivel con el que se pueden escribir tanto programas convencionales como para Internet. Java incluye dos elementos: un compilador y un intérprete. El compilador (programa traductor) produce un código de bytes que se almacena en un fichero para ser ejecutado por el intérprete Java denominado máquina virtual de Java.

Java es un lenguaje de programación más utilizado para escribir programas de código abierto la misma que se puede escribir el código con mucha tibieza y gracia a los programas lenguaje orientado a objetos se puede eliminar muchos errores.

## **2.9 Entornos de Desarrollo**

En la actualidad existen una multitud de fabricantes que disponen de entornos de desarrollo para Java. Las mismas que están teniendo una gran aceptación en algunos entornos de libre distribución, Puede utilizar otros entornos, aunque sean de pago, ya que existen versiones reducidas que se pueden utilizar para aprender los fundamentos del lenguaje y la programación con Java (Allende, 2005).

### **2.9.1 NetBeans**

NetBeans es un proyecto de creación de un entorno de libre distribución profesional patrocinado por Sun Microsystems. En [java.sun.com](http://java.sun.com) puede encontrar un paquete completo que

Viene con el entorno de desarrollo NetBeans y la última versión del lenguaje Puede encontrarlo en NetBeans.

### **2.10 Metodologías Agiles**

Según (Sevilla, 2016) dice las metodologías ágiles son metodologías con una filosofía de trabajo alternativo a las metodologías tradicionales. Estas metodologías aplican procesos que desarrollan el software de manera iterativo e incremental, con enfoque distinto, un enfoque basado en el producto.

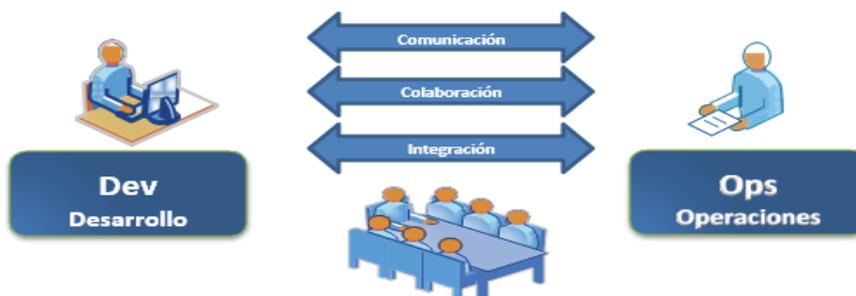
### **2.11 DevOps**

Con la llegada de las metodologías ágiles de desarrollo y las necesidades de realizar integración y entrega continua (CI, continuous integration y CD, continuous delivery) aparece una nueva corriente organizativa llamada DevOps que, en resumidas cuentas pretende aunar en un único equipo a perfiles muy separados en organizaciones más tradicionales como puedan serlos desarrolladores y los equipos de operaciones, todo ello con el objetivo final de realizar despliegues en entornos productivos de forma más regular. Haciendo nuevas entregas del software de una forma regular (semanalmente, diariamente o, incluso, varias veces al día) se consigue dotar al proceso del paso a producción de más seguridad o estabilidad y más eficiencia ( Rodríguez, 2018)

En la Figura 13 se puede observar la integración, colaboración y comunicación entre el equipo de desarrollo y el equipo de operaciones.

**Figura 2**

*Aplicando la metodología DevOps*



Nota: El grafico nos demuestra la aplicación real de la metodología en los departamentos de desarrollo y operaciones. tomado de (Karina, 2017)

### 2.12 Fases de DevOps

Cada fase debe contribuir no solo al área de tecnología sino a la organización en general para el alcance de los objetivos con respecto a la entrega de software y lograr un equilibrio entre el costo, velocidad y calidad de las aplicaciones. Se presentan las fases según (Kort, 2016):

#### **Planificación.**

En la fase de *Planificación* es donde se recolectan todos los requerimientos, aquí se realizan la creación de las tareas y todas las actividades del proyecto.

#### **Codificación.**

La fase de *Codificación* se da con el desarrollo o actualización del código de acuerdo a la tarea planificada. Hay que considerar las herramientas que ayudan a mejorar el rendimiento de esta fase en donde se debe tener un eficaz control de versiones.

**Construcción.**

En esta fase se realiza la Construcción del código con su respectiva compilación, se realiza el análisis estático del código y empieza la ejecución de las pruebas unitarias en caso de presentarse algún error se procede a notificar los fallos a los responsables respectivos.

**Pruebas.**

En la fase de *Pruebas* el código se revisa antes del despliegue en Producción. Automatizar las pruebas implica el uso de herramientas adecuadas que faciliten realizar las pruebas por separado para la creación de las secuencias de los comandos que se ejecuten continuamente sin intervención manual alguna.

**Despliegue.**

En la fase de *Despliegue* se configuran las herramientas para que el despliegue de las aplicaciones sea en forma automatizada. En este trabajo se dará a través del despliegue de los microservicios en los contenedores

**Monitoreo.**

En la fase de Monitoreo se supervisa el funcionamiento de las aplicaciones desplegadas, esto ayuda a detectar posibles fallos que no se hayan controlado en las fases anteriores. Con un continuo seguimiento se puede medir la disponibilidad y el rendimiento del producto brindando así una mayor estabilidad del ambiente en tiempo real. Identificar cada una de las fases de DevOps es esencial para adoptar la cultura en las organizaciones (Karina, 2017)

## CAPITULO III

### 3. INVESTIGACIÓN E IMPLEMENTACIÓN

#### 3.1. Introducción

En el presente trabajo se enfoca a la automatización del proceso de desarrollo de software mediante una herramienta de integración continua (Jenkins) para el control de versionamiento en la Dirección de Tecnologías de la información y Comunicaciones del Ejercito. De tal forma que para el desarrollo de la integración continua utilizará la metodología DevOps, y se emplean diferentes fases como son: Planificación, codificación, construcción, pruebas, despliegue.

La integración continua con la herramienta Jenkins está encaminado al control de versionamiento de tal manera que los desarrolladores podrán revisar las métricas de calidad del software establecidas por el equipo de trabajo, enviar las modificaciones del software, una vez pasadas todas las validaciones, al repositorio principal, automatizar la compilación del software o su despliegue, una vez se hayan integrado nuevos cambios en el proyecto, notificar debidamente a los desarrolladores o al equipo de aseguramiento de la calidad cuando se encuentra cualquier tipo de error, ya sea en base a las pruebas del software o a las métricas de calidad.

A continuación, se presenta el proceso de desarrollo de software, aplicando la metodología DevOps.

#### 3.2. Recomendaciones de hardware para instalación de Jenkins

Es de muy importancia la configuración por primera vez una instancia de Jenkins de forma adecuada es el hecho de que no existe un tamaño que se adapte a todos los casos de uso. La

especificación exacta de hardware que necesitarás estará muy estrechamente ligada a las necesidades presentes y futuras de tu organización.

El nodo maestro de Jenkins servirá peticiones HTTP y almacenará toda la información de tu instancia de Jenkins en su directorio \$JENKINS\_HOME: configuraciones, históricos de ejecuciones, plugins instalados, etc.). Es importante tener en cuenta además que aquellos nodos maestros que se ejecuten en alta disponibilidad usando el plugin de CloudBees “High Availability plugin” tendrán que instalar el directorio \$JENKINS\_HOME en un servidor de datos de la red, el cual será accesible a los servidores de Jenkins por NFS o algún otro protocolo de compartición de archivo (De la Peña, 2015).

### **3.3. Requerimientos básicos para instalación de Jenkins**

Java 8

256MB de memoria RAM

1GB+ de espacio en disco duro

Cabe mencionar que estos requerimientos dependen mucho de lo que planeamos hacer con Jenkins. Estos requerimientos son los mínimos y es recomendable ajustarlos conforme el equipo y el número de usuarios que envían hacen uso del mismo crece. Adicionalmente es requerido que Jenkins tenga suficiente ancho de banda para funcionar ya que estará descargando toneladas de código (Geek, 2018)

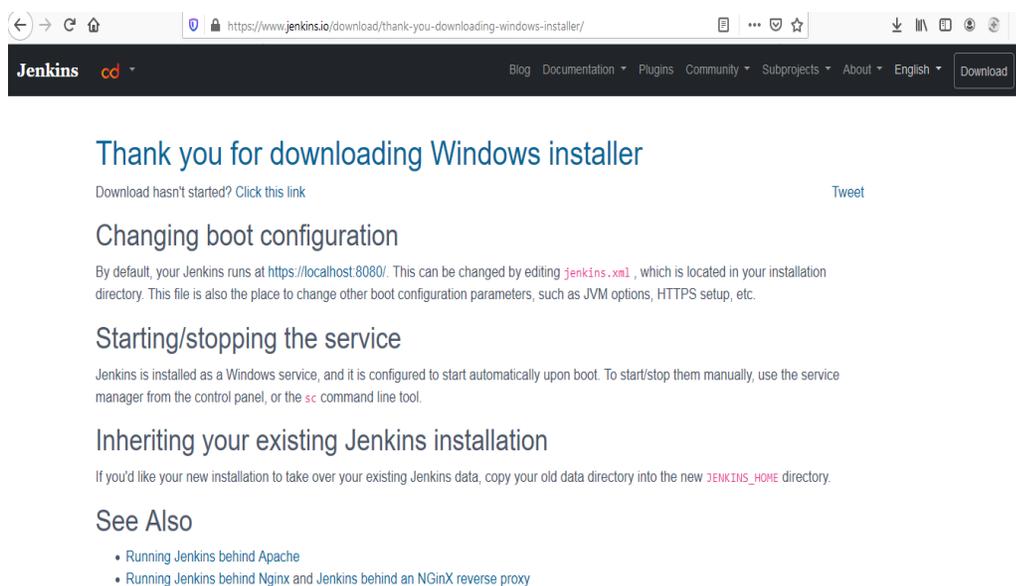
### **3.4. Instalar y configuración Jenkins en Windows server 2016**

Para poder instalar Jenkins en Windows server 2016 es necesario que se instálale Jdk 8 para que la instalación sea de mejor manera y su funcionamiento sea eficaz.

1.- Realizar la descarga del instalador debemos dirigirnos al sitio web oficial <https://www.jenkins.io/> para la descarga de la herramienta de integración continua damos clic en descargar.

### Figura 3

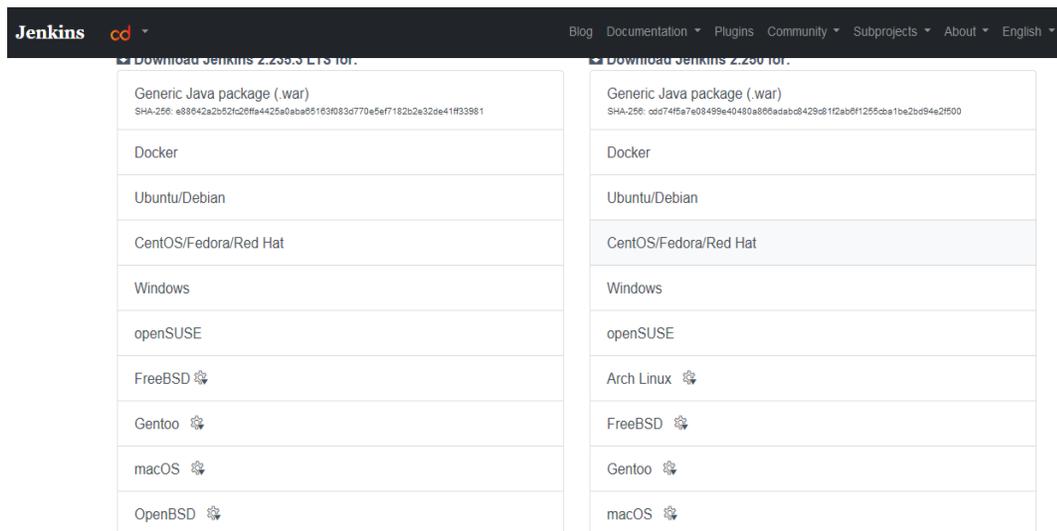
*Página oficial para la descarga Jenkins*



Nota: El gráfico indica el sitio web oficial para la descarga de Jenkins.

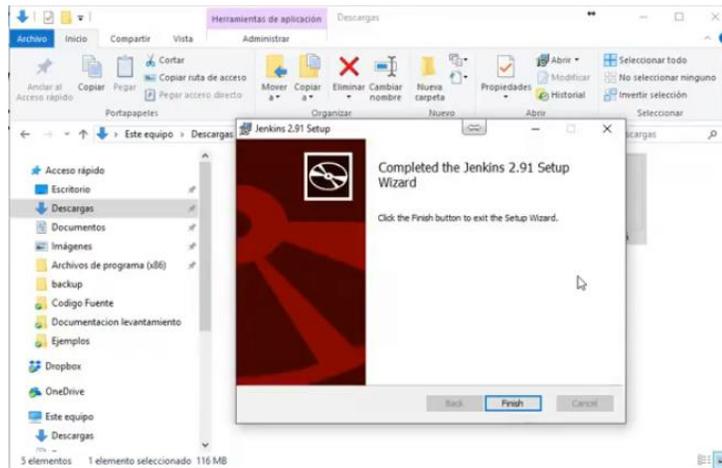
Una vez ya cuando nos encontramos en el sitio oficial debemos de realizar la descarga de la herramienta tomando en cuenta para que sistema operativo lo vamos a necesitar Jenkins puede ser para Linux o Windows server 2016, en este caso lo instalaremos para Windows server 2016.

Figura 4

*Selección de sistema Operativo*

Nota: El gráfico indica la descarga de Jenkins para Windows server 2016.

Terminado la descarga realizamos la instalación en el sistema operativo la cual se instalará en el disco local C en la carpeta system 32 damos clic en siguiente.

**Figura 5***Instalación de Jenkins*

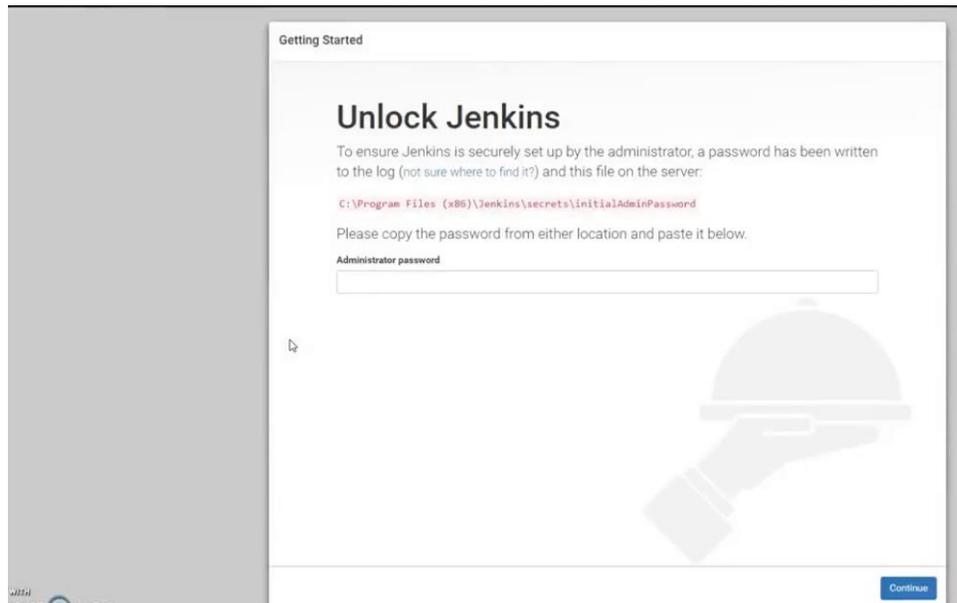
Nota: El gráfico indica la finalización de la instalación Jenkins

Para continuar con la instalación nos pedirá una contraseña para poder seguir la cual lo vamos a esta dirección en el disco:

C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins.jenkins una vez conseguida la copiamos y pegamos y damos clic en continua.

**Figura 6**

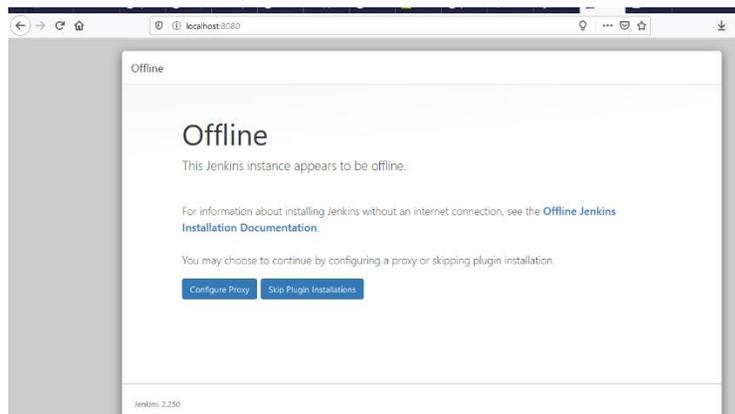
*Contraseña para la instalación*



Nota: El gráfico indica el ingreso de contraseña para la instalación.

Abre tu navegador con la IP o dominio de tu servidor en el puerto 8080

(<http://dominioservidor.com:8080>) y deberás ver lo siguiente en la cual debemos descargar plugins.

**Figura 7***Instalación de Jenkins*

Nota: El gráfico indica la instalación de Jenkins.

Ya finalizado la instalación ahora sí podremos dar de alta nuestro usuario para ingresar a la interfaz web. Debemos crear un usuario y una contraseña para la administración de la herramienta de integración continua Jenkins.

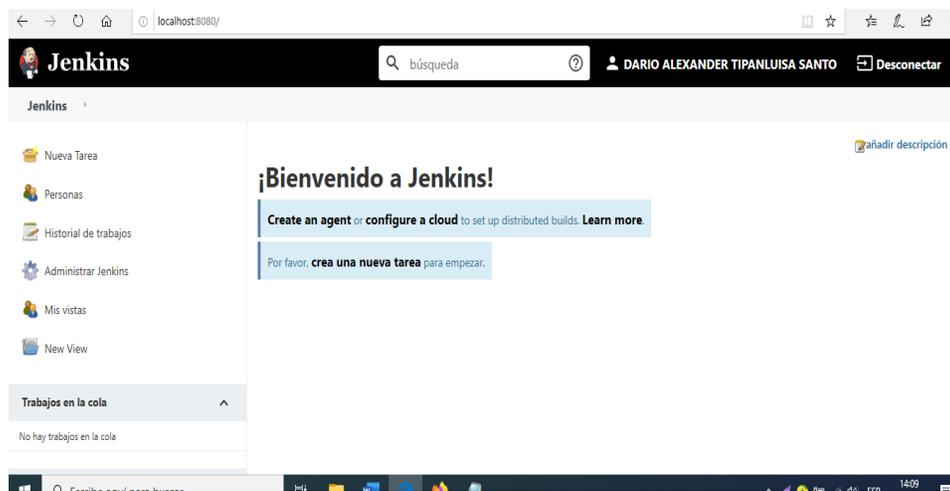
**Figura 8***Creación de usuario y contraseña*A screenshot of the Jenkins 'Getting Started' page, specifically the 'Create First Admin User' section. The form contains five input fields: 'Usuario:', 'Contraseña:', 'Confirma la contraseña:', 'Nombre completo:', and 'Dirección de email:'. At the bottom of the form, there are two buttons: 'Continue as admin' and 'Save and Finish'. The footer of the page shows 'Jenkins 2.89.3'.

Nota: El gráfico indica creación de usuario y contraseña de Jenkins.

Una vez ya creado el usuario y contraseña ya podemos ingresar al servidor y podemos visualizar esta interfaz gráfica en la cual tenemos opciones las cuales es necesaria para realizar la integración continua

## Figura 9

*Finaliza la instalación Jenkins Windows server*



Nota: El grafico indica la instalación completa del servidor de automatización Jenkins.

### 3.7. Instalación de subversión con tortoise svn

Para poder integrar Jenkins con el repositorio svn es necesario realizar los pasos que a continuación vamos a tratar para cual podemos hacer es la descarga del servidor una vez ya descargada podemos realizar la instalación ya una vez instalado debemos subir nuestros proyectos al repositorio.

**Figura 10**

*Instalación de TortoiseSVN*

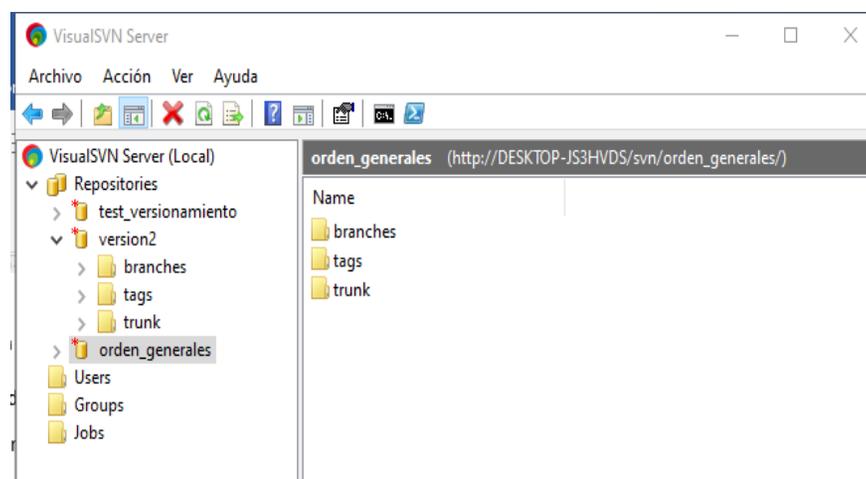


Nota: El grafico indica la descarga de tortoise SVN para creación de repositorios

Pero antes debemos de descargar TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones Apache Subversión. Esto significa que TortoiseSVN administra archivos y directorios a lo largo del tiempo y así poder subir los proyectos al repositorio.

**Figura 11**

*Interfaz gráfica de SVN*



Nota: El gráfico indica la interfaz donde se pueden subir los proyectos.

### 3.8. Creación de repositorio

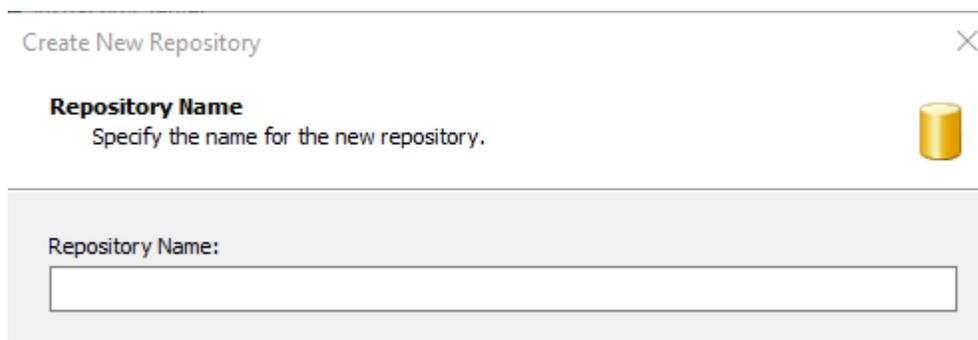
La creación de repositorio subversión permite a los desarrolladores ingresar al repositorio a través de redes, lo que le permite ser usado por desarrolladores que se encuentran en computadoras y realizando el mismo proyecto o a su vez varios y trabajando en distintos sistemas operativos. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración

Pasos para crear el repositorio

- 1.- ingresamos al servidor y damos clic crear nuevo repositorio
- 2.- le damos un nombre del repositorio la cual podremos identificarlo le damos siguiente

#### Figura 12

*Creación de repositorios*



Create New Repository

**Repository Name**  
Specify the name for the new repository.

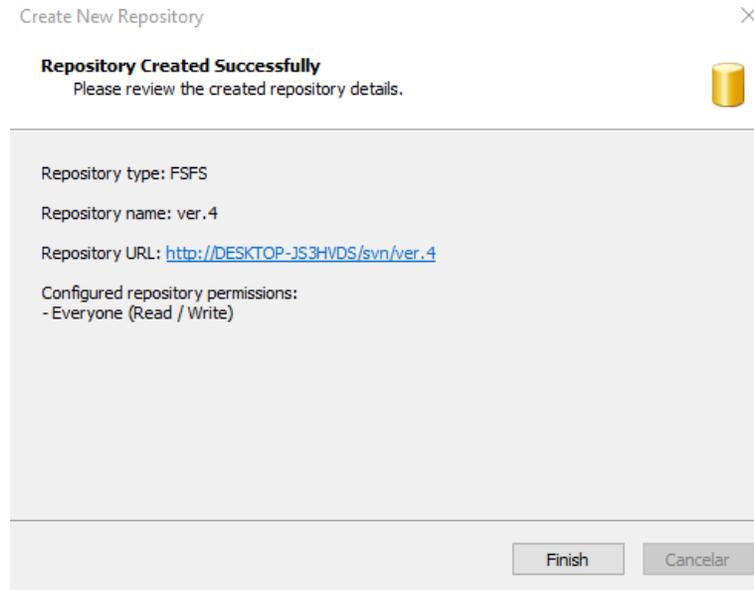
Repository Name:

Nota: El gráfico indica la creación del nombre del repositorio.

3.- Al finalizar la creación del repositorio al finalizar la creación nos dará un link donde está ubicado el repositorio la cual nos va a servir para poder realizar la integración continua con el servidor de automatización (Jenkins)

**Figura 13**

*Dirección del repositorio*



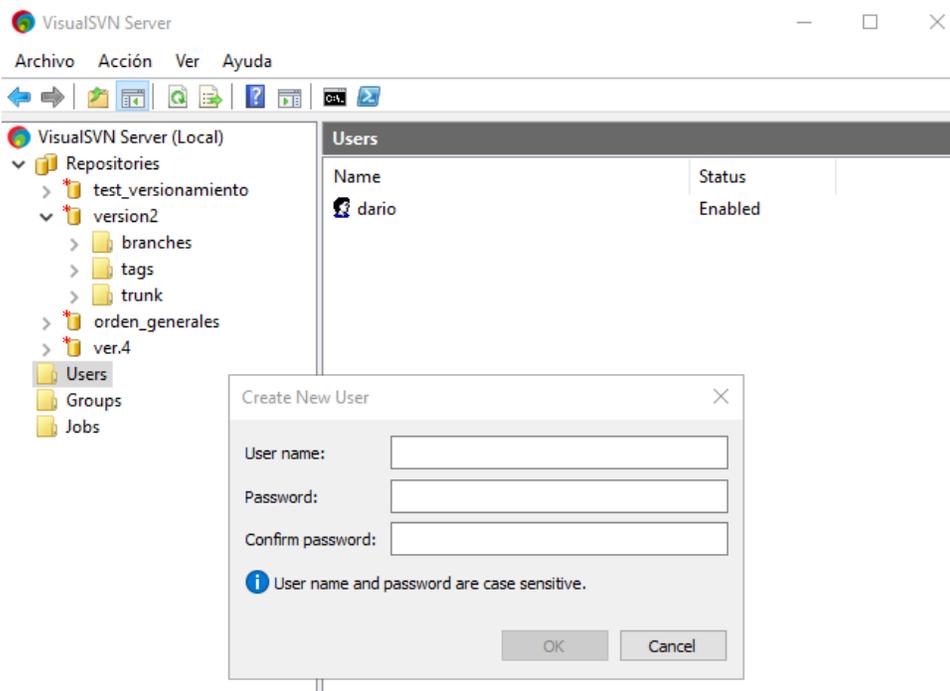
Nota: El gráfico indica la dirección exacta donde está el repositorio.

### 3.9. Creación de usuario y subida de proyectos al repositorio

La creación de usuario de conexión la cual nos permitirá conectar con nuestro servidor de automatización y así poder tener las credenciales necesarias para que se puedan conectar.

Pasos para crear el usuario y la contraseña

- 1.- ingresamos al repositorio
- 2.- damos clic derecho en crear usuario
- 3.- ingresamos usuario y contraseña y damos ok.

**Figura 14***Creación de usuario*

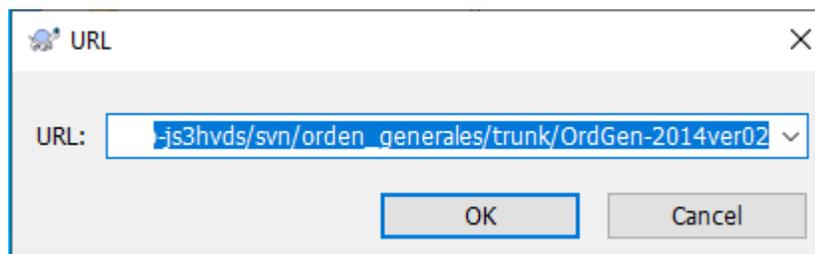
Nota: El gráfico indica la creación de usuario y contraseña.

Para agregar proyectos a nuestro repositorio ya una vez creado y configurado ya tenemos que comenzar a usarlo para lo cual debemos realizar los siguientes pasos que a continuación vamos a realizar.

1.- Damos clic derecho en el escritorio para luego poder ingresar en tortoise svn y así nos de la dirección para poder dar ruta de nuestro repositorio

Figura 15

*Dirección de repositorio*



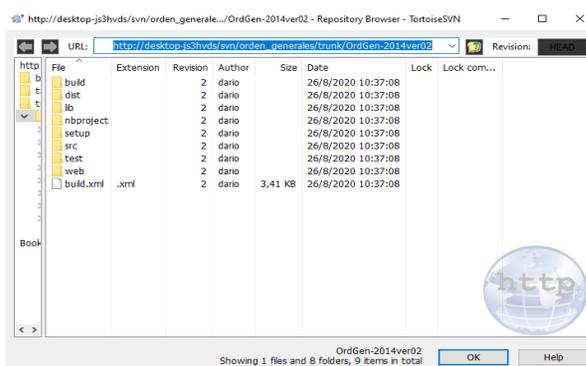
Nota: El gráfico indica la dirección por donde está ubicado el código fuente del proyecto en el repositorio subversión.

2.-Una vez ya en nuestro repositorio nos pedirá usuario y contraseña para lo cual debemos recordar la que creamos en el usuario

3.- luego vamos a arrastrar r nuestro proyecto y poder subirlos a nuestro repositorio para poder integrarlos con el servidor de automatización

Figura 16

*Subida de proyecto*



Nota: El gráfico indica que ya está subido el proyecto en el repositorio subversión.

### 3.10. Integración de Jenkins y subversión

Para realizar la integración de Jenkins con subversión debemos tener en cuenta si está instalado el plugin (complemento) y así podamos hacer la integración una vez verificado realizamos las configuraciones para que el proyecto pueda realizarse de la mejor manera

1.- Creamos un nuevo proyecto de estilo libre la cual nos permitirá realizar las configuraciones que vamos a necesitar para poder integrar.

#### Figura 17

##### Creación de proyecto

The screenshot shows the Jenkins web interface for creating a new project. The main heading is "Enter an item name". A text input field contains the text "version 3". Below the input field, there is a small red message that says "» Required field".

Below this, there is a section titled "Crear un proyecto de estilo libre" (Create a free-style project). It includes a brief description: "Esta es la característica principal de Jenkins. la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización."

Underneath, there is a section titled "If you want to create a new item from other existing, you can use this option:". It features a "Copy from" dropdown menu and a text input field with the placeholder "Type to autocomplete".

At the bottom of the form, there is a prominent blue "OK" button.

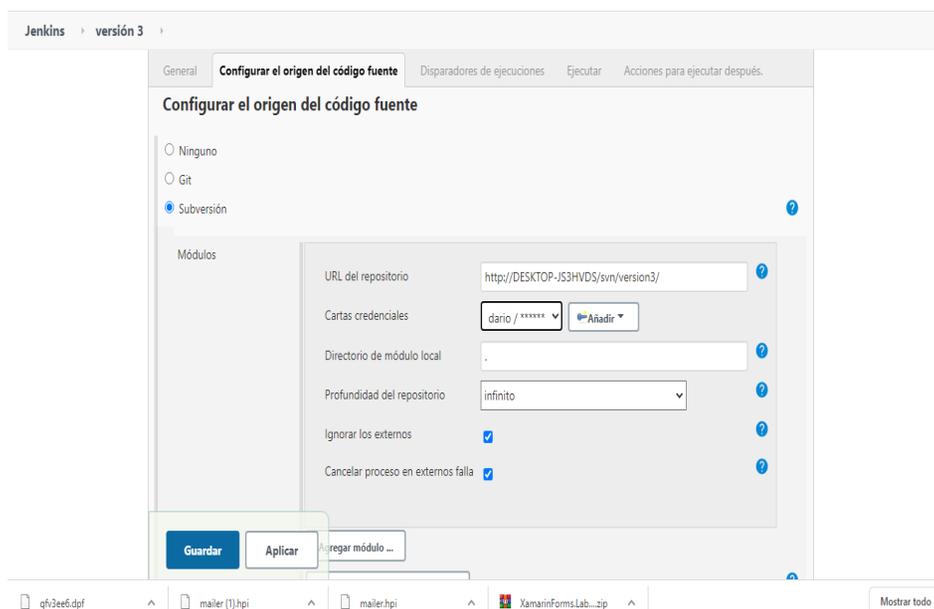
The top of the page shows the Jenkins logo, a search bar, and the user profile "DARIO ALEXANDER TIPANLUISA SANTO" with a "Desconectar" (Disconnect) button. The breadcrumb navigation shows "Jenkins > Todo >".

Nota: El gráfico indica creación de proyecto de integración continua mediante el servidor de automatización.

Luego de crear el proyecto nos dirigimos a las configuraciones la cual nos dirigimos a configuraciones de origen del código fuente para allí ingresar la ruta del repositorio la dirección de ruta de la conexión del repositorio y las credenciales que nos den permiso de conexión

**Figura 18**

### *Configuración de Jenkins con subversión*



Nota: El gráfico indica la creación del origen del código fuente dando la dirección del repositorio y así realizar la integración.

3.- Para que se ejecuten los disparos debemos configurar para que las consultas de repositorio y le configuramos para que tiempo necesita que el servidor de automatización realiza las consulta para ver si hay algún cambio y así se pueda actualizar una vez realizado toda la configuración guardamos los cambios y salimos.

Figura 19

*Configuración de disparos de ejecución*

**Disparadores de ejecuciones**

- Lanzar ejecuciones remotas (ejem: desde 'scripts')
- Construir tras otros proyectos
- Consultar repositorio (SCM)

Programador:

Would last have run at miércoles 26 de agosto de 2020 18H00' COT; would next run at miércoles 26 de agosto de 2020 18H15' COT.

- Ignore post-commit hooks
- Ejecutar periódicamente
- GitHub hook trigger for GITScm polling

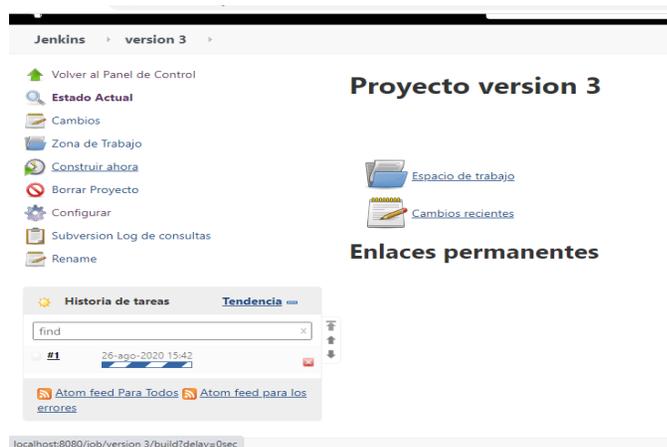
**Guardar** **Apply**

Nota: El gráfico indica la configuración los disparos de la ejecución en el repositorio cada cierto tiempo.

3.- Podemos realizar la ejecución de la tarea dando clic en construir ahora la cual realizará el testeo de todo el código del repositorio la misma que una vez finalizado nos dará que el proceso terminado con éxito.

Figura 20

construcción del proyecto



Nota: El gráfico indica la construcción de proyecto en servidor.

Finalizado la ejecución nos dirigimos a consola dónde podemos ver los detalles del testeo que se conectó Jenkins y el repositorio subversión.

Figura 21

Verificación de ejecución



Nota: El gráfico indica la salida de consola la cual se ejecutó el proceso de verificación de código fuente en el repositorio.

## CAPITULO IV

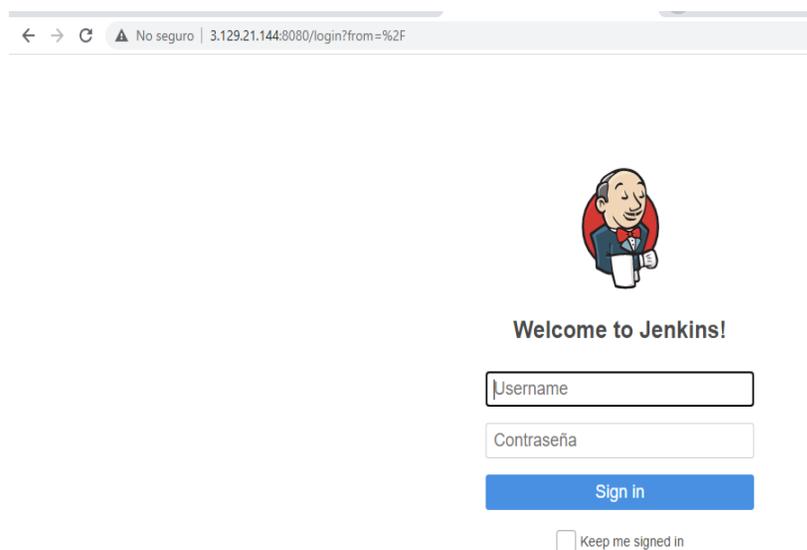
### 4. FUNCIONAMIENTO

#### 4.1. Verificación del funcionamiento del servidor de automatización Jenkins

En esta parte del capítulo se presenta la fase de funcionamiento en el servidor de pruebas de la Tics Del ejército ecuatoriano donde se conoce que la integración del servidor de automatización mediante la cual podemos ingresar al servidor de cualquier parte del mundo mediante una conexión del servicio de internet y digitando la ip y el respectivo puerto en este caso podemos ingresar de la siguiente manera, digitando en cualquier navegador la siguiente ip 3.129.21.144:8080

#### Figura 22

*Ingreso al servidor*



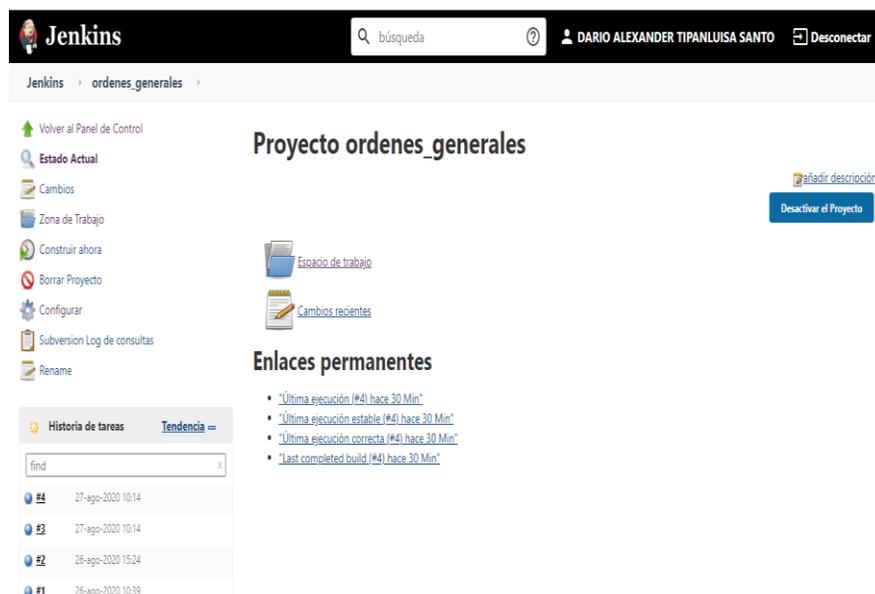
Nota: El gráfico nos muestra el funcionamiento del servidor se puedo abrir desde cualquier parte del mundo y así poder administrarlo de mejor manera.

## 4.2. Control del historial de tareas

El historial de tareas nos permite verificar las tareas y el funcionamiento automatizado del servidor está funcionando realizando una consulta al repositorio y verificando si existe algún cambio y su normal funcionamiento conectado subversión se estabiliza y realizan las pruebas del producto, teniendo en cuenta que sea de óptima calidad y logre cumplir el ciclo de desarrollo.

**Figura 23**

*Consultas al repositorio*



The screenshot displays the Jenkins web interface for the 'ordenes\_generales' project. The 'Historia de tareas' (Task History) section is visible, showing a table of task entries. The table has columns for task status (represented by icons) and timestamps. The entries are as follows:

Icono	Fecha y Hora
🔵	27-ago-2020 10:14
🔵	27-ago-2020 10:14
🔵	26-ago-2020 15:24
🔵	26-ago-2020 10:39

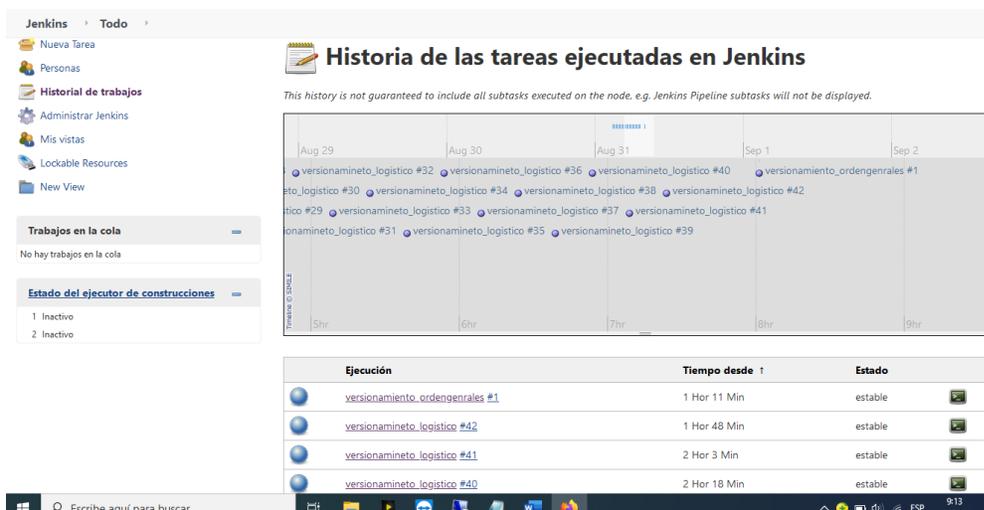
Nota: El gráfico nos indica las consultas del servidor de automatización con el repositorio subversión Elaborado por autor.

Verificamos la historia de trabajos de consultas al repositorio que se está realizando y así poder tener control de los proyectos y encontrar los errores de una forma mucho más rápida.

Al dar clic en construir ahora, podemos ver el proceso de automatización que realiza y nos muestra lo que hace el servidor con los proyectos de repositorio. Verificar su funcionamiento que se actualiza cada cierto tiempo según la configuración del servidor en este caso lo realiza cada 15 minutos o ya depende del administrador el tiempo que desea que el servidor. Demos tener el control de su funcionamiento ingresando al historial de las tareas de ejecución. Observamos si existen alguna falla o si algo no funciona bien y su estabilidad de funcionamiento por fechas.

**Figura 24**

### Ejecución



Nota: El gráfico nos indica la funcionalidad del servidor de automatización que se encuentra estable sin ninguna falla o cambio.

## CAPITULO V

### 5. CONCLUSIONES Y RECOMENDACIONES

#### 5.1 Conclusiones

- Se realizo el análisis de entorno de desarrollo del servidor de automatización para lo cual se pudo analizar para que sirve cada uno de ellos, donde pueden gestionar los usuarios, mostrar las tareas y los procesos, adicionalmente se puede realizar a la conexión mediante complementos que pueden extender las funcionalidades de Jenkins para una mejor optimización.
- La metodología DevOps es la que mejor se adapta a la integración continua porque permite el trabajo entre los departamentos de desarrollo y operaciones permitiendo compartir archivos y la automatización de procesos por lo cual facilita la integración.
- Las pruebas de integración continua con la herramienta Jenkins permite la verificación con las actualizaciones o modificaciones que realizan el departamento de desarrollo u el departamento de operaciones los cuales se conecta a través del repositorio subversión.

## 5.2 Recomendaciones

- Realizar un análisis con que arquitectura tecnológica contamos o cuentan las empresas para poder implementar el servidor de automatización que permita la integración continua para el control de los proyectos que están en producción.
- Entender cómo aplicar correctamente la metodología DevOps para que todas sus fases: planificación, codificación, construcción, pruebas, despliegue y monitoreo se puedan ejecutar de la manera correcta y eficaz.
- Realizar pruebas de integración continua reiteradamente para verificar si el servidor de automatización está controlando los versionamientos de los proyectos antes de ser implementados en producción.

**REFERENCIAS BIBLIOGRÁFICAS**

Rodríguez, Á. O. (2018). *DevOps y análisis de performance automático*. Universidad Oberta de Catalunya .

Allende, J. S. (2005). *Programación en Java 2* . McGraw-Hill España.

Álvarez, A. R. (2005). *INTRODUCCIÓN A LA AUTOMATIZACIÓN DE PROCESOS*. Universidad de oviedo .

ÁLVAREZ, R. B., & CANTÓN MAYO , I. (2009). *Las tecnologías de la información y la comunicación*. Universidad de Leon .

Arcis, F. (2017). *Diseño e Implementacion de un sistema de integracion y entrega continua con Jenkins*. Universidad tecnica Editorial .

Avantaggiato, M. A., Castro, A. D., & Loiacono, P. (2013). *Jenkins Integración continua*. Universidad Nacional de Quilmes.

Bersabé, J. E. (2016). *Integración y Despliegue Continuo: Monitorización de y. sevilla*: Universidad de Sevilla.

Borrell, G. ( 2006). *El control de versiones*. Copyright ©.

De la Peña, M. (8 de julio de 2015). *mdelapenya*. Recuperado el 30 de julio de 2020, de <https://mdelapenya.github.io/jenkins/recomendaciones-hardware-jenkins/>

Ecuatoriano, e. (2020). *ejercito ecuatoriano marca la victoria* . Obtenido de <https://ejercitoecuadoriano.mil.ec/institucion/fftt/sistema-de-armas/comunicaciones>

Fernandez, J. (2017). *Implementacion de un sistema de integracion continua en una metodologia consolidada*. Universidad de Castilla La Mancha .

- Geek, R. (3 de julio de 2018). *ricardogeek*. Recuperado el 30 de julio de 2020, de <https://ricardogeek.com/guia-para-instalar-jenkins-desde-la-terminal/>
- Izaguirre, E. (2012). *Sistema de Automatizacion*. Feijóo,.
- Karina, F. A. (2017). *Definición de un ambiente de construcción de aplicaciones empresariales a través de Devops, Microservicios y Contenedores*. salinas : Universidad particular de Loja.
- OROZCO, A. M. (2015). *LA INTEGRACIÓN CONTINUA Y SU APORTE AL ASEGURAMIENTO DE LA CALIDAD EN EL CICLO DE VIDA DEL DESARROLLO DE SOFTWARE*. BOGOTÁ: UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS.
- PABLO, D. B., USAMENTIAGA FERNANDEZ , R., & BARREGO , P. (2019). *ESTUDIO MEJORA E IMPLEMENTACION DE UN SISTEMA DE INTEGRACION CONTINUA*. UNIVERSIDAD OBIEDO.
- Sánchez Duarte, E. (2008). *LAS TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN (TIC) DESDE UNA PERSPECTIVA*. Costa Rica : Universidad Nacional.
- Schwarber , K., & Sutherland, J. (2013). *La Guía de Scrum - La Guía Definitiva de Scrum: Las Reglas del Juego*.
- Sevilla, A. M. (2016). *Configuración de un entorno de trabajo para el Desarrollo de software agil con integracion continua* . Universidad Politecnica de Madrid .
- Sierra, F. J. (2015). *Java2: Lenguaje y aplicaciones*. RA-MA.