

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE SERVICIOS DE RED CON HERRAMIENTAS DE DEVOPS

DESPLIEGUE DE WEB HOSTING CONTROL PANEL SOFTWARE MEDIANTE CONTENEDORES

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN REDES Y TELECOMUNICACIONES**

ADRIANA SAMANTA CHALAVARRI PIARPUEZAN

adriana.chalavarri@epn.edu.ec

DIRECTOR: FERNANDO VINICIO BECERRA CAMACHO

fernando.becerrac@epn.edu.ec

DMQ, agosto 2023

CERTIFICACIONES

Yo, ADRIANA SAMANTA CHALAVARRI PIARPUEZAN declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

ADRIANA SAMANTA CHALAVARRI PIARPUEZAN

adriana.chalavarri@epn.edu.ec

adrianachalavarri@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por ADRIANA SAMANTA CHALAVARRI PIARPUEZAN, bajo mi supervisión.

FERNANDO VINICIO BECERRA CAMACHO
DIRECTOR

fernando.becerrac@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

ADRIANA SAMANTA CHALAVARRI PIARPUEZAN

DEDICATORIA

A mi madre, por su apoyo, paciencia, esfuerzo y amor inquebrantable.

Adriana Samanta

AGRADECIMIENTO

Un agradecimiento especial a Olga Mariela y Alba Yolanda, por apoyarme y quererme de la forma en que lo hacen, por estar presentes en cada etapa de mi vida, pero, sobre todo, por nunca dejar de creer en mí.

A Estefany, Gabriel y Mathías, gracias por estar siempre a mí lado y darme ánimos para afrontar todas mis dificultades.

A Christopher Vizcaíno y a Hugo Guepud, por estar siempre presentes.

A mi tutor, Ing. Fernando Becerra, por ayudarme y guiarme durante mi etapa académica en la ESFOT.

Al Ing. Rodrigo Luna miembro de la Dirección de Gestión de la Información y Procesos de la EPN, por ser un excelente mentor y amigo, gracias por sus enseñanzas y consejos.

A mis amigos, Geomaira, Stalin y Byron, por su apoyo incondicional.

Adriana Samanta

ÍNDICE DE CONTENIDOS

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDOS	V
RESUMEN	VII
<i>ABSTRACT</i>	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	9
1.1 OBJETIVO GENERAL	9
1.2 OBJETIVOS ESPECÍFICOS	9
1.3 ALCANCE	10
1.4 MARCO TEÓRICO.....	10
DEVOPS.....	10
<i>WEB HOSTING CONTROL PANEL SOFTWARE</i>	11
<i>DOCKER</i>	11
2 METODOLOGÍA	12
3 RESULTADOS	14
3.1 ANÁLISIS DE LA HERRAMIENTA DE DEVOPS	14
CONTENEDORES	15
<i>DOCKER</i>	17
<i>WEB HOSTING Y WEB HOSTING CONTROL PANEL SOFTWARE</i>	19
<i>WEB HOSTING CONTROL PANEL SOFTWARE</i>	19
<i>CPANEL</i>	20
<i>WEBMIN</i>	21

3.2	DISEÑAR LA SOLUCIÓN PARA CADA SERVICIO DE <i>NETWORKING</i> MEDIANTE HERRAMIENTAS DE DEVOPS	23
	INSTALACIÓN DE <i>SOFTWARES</i> PARA LA EJECUCIÓN DE <i>WEBMIN</i> EN CONTENEDORES	23
	HERRAMIENTAS Y DEPENDENCIAS PARA LA IMPLEMENTACIÓN DE <i>WEBMIN</i>	24
3.3	IMPLEMENTAR LAS SOLUCIONES MEDIANTE HERRAMIENTAS DE DEVOPS PARA EL DESPLIEGUE DE LOS SERVICIOS DE <i>NETWORKING</i>	25
	ESTRUCTURA DE LA IMAGEN <i>WEBMIN</i>	25
3.4	VERIFICACIÓN DEL FUNCIONAMIENTO DE CADA SERVICIO DE <i>NETWORKING</i> IMPLEMENTADO MEDIANTE DEVOPS	30
4	CONCLUSIONES	38
5	RECOMENDACIONES	39
6	REFERENCIAS BIBLIOGRÁFICAS	40
7	ANEXOS	45
	ANEXO I: Certificado de Originalidad	45
	ANEXO II: Código QR de la implementación y pruebas de funcionamiento	i
	ANEXO III: Códigos Fuente.....	ii

RESUMEN

DevOps permite la entrega segura, rápida, continua y confiable de *software*, contribuyendo a la escalabilidad de los servicios de *networking* con el uso de tres pilares claves: automatización, colaboración, y la mejora continua.

En este contexto, el presente trabajo de titulación se encuentra estructurado en cuatro secciones que proporcionan información sobre la implementación y gestión de *web hosting control panel software*.

La primera sección del documento complementa la información proporcionada en el marco teórico sobre DevOps, *Docker*, *web hosting* y *web hosting control panel software*.

En base a la información presentada en la primera sección, se establecerán pautas para la elaboración del contenedor en base al desarrollo de la imagen del panel de control, *Webmin*. En el proceso se va a considerar el sistema operativo en el cual se va a basar la nueva imagen y todas las dependencias que se requieran implementar para garantizar su construcción y despliegue.

En la tercera sección del documento, se desarrollarán las pautas establecidas previamente, y con ello el entorno de aplicación e implementación de instrucciones coherentes con el sistema operativo y se detallará la estructura de la imagen creada en *Dockerfile* para ejecutar el *software*.

Por último, en la sección final del documento se realizarán pruebas de funcionamiento del *software* (*Webmin*), y su respectiva configuración para el acceso a la plataforma, a través del navegador.

De presentar fallas, se buscará una solución que permita solventar el inconveniente para obtener el resultado esperado.

PALABRAS CLAVE: DevOps, *web hosting control panel software*, *Docker*, *Dockerfile*, *Webmin*.

ABSTRACT

DevOps enables the secure, fast, continuous, and reliable delivery of software, contributing to the scalability of networking services with the use of three key pillars: automation, collaboration, and continuous improvement.

In this context, the present degree work is structured in four sections that provide information on the implementation and management of web hosting control panel software.

The first section of the paper complements the information provided in the theoretical framework on DevOps, Docker, web hosting and web hosting control panel software.

Based on the information presented in the first section, guidelines will be established for the elaboration of the container based on the development of the control panel image, Webmin. The process will consider the operating system on which the new image will be based and all the dependencies that need to be implemented to ensure its construction and deployment.

In the third section of the document, the previously established guidelines will be developed, and with it the application environment and implementation of instructions consistent with the operating system and the structure of the image created in Dockerfile to run the software will be detailed.

Finally, in the final section of the document, the software (Webmin) and its respective configuration for accessing the platform through the browser will be tested.

In case of failures, a solution will be sought to solve the problem to obtain the expected result.

KEYWORDS: *DevOps, web hosting control panel software, Docker, Dockerfile, Webmin.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Para llevar a cabo el desarrollo del proyecto, se presenta una investigación acerca de *web hosting control panel software* y las aplicaciones que brindan este servicio, con el objetivo de conocer los beneficios que la herramienta ofrece. Asimismo, se proporciona información detallada que contribuye al entendimiento de la plataforma tecnológica *Docker* y su herramienta *Dockerfile*, para la elaboración de imágenes en contenedores.

En base a los conocimientos obtenidos previamente, se establece un plan que permite comprender como se va a ejecutar el *software* de panel de control en el contenedor. También, se detallan las instrucciones definidas en el archivo *Dockerfile* para la construcción de la imagen y las configuraciones para ejecutar el contenedor.

Además, se van a realizar pruebas de funcionamiento y configuraciones adicionales para garantizar el despliegue exitoso del *web hosting control panel software*, y evidenciar su funcionamiento. En definitiva, al coordinar las capacidades de las herramientas mencionadas, se ofrece una solución efectiva para la automatización de los servicios de *networking*.

1.1 OBJETIVO GENERAL

Despliegue de *web hosting control panel software* mediante contenedores.

1.2 OBJETIVOS ESPECÍFICOS

- Analizar la herramienta de DevOps.
- Diseñar la solución para cada servicio de *networking* mediante servicios de DevOps.
- Implementar las soluciones mediante herramientas de DevOps para el despliegue de los servicios de *networking*.
- Verificar el funcionamiento de cada servicio de *networking* implementado mediante DevOps.

1.3 ALCANCE

El presente proyecto tiene como objetivo estudiar y analizar las herramientas de DevOps y los beneficios en la gestión de operaciones y desarrollo de *software*, para implementar una solución que permita gestionar y administrar los diferentes servicios de *networking* de manera que, automatice procesos y tareas sin requerir de una intervención directa.

1.4 MARCO TEÓRICO

DEVOPS

DevOps es una filosofía tecnológica en constante evolución, que proviene de los términos de desarrollo de *software* (*Dev*) y operaciones de TI (*Ops*), cuyo propósito es promover la entrega de *software* de alta calidad de forma rápida, eficiente, segura e innovadora para el ámbito empresarial. DevOps está constituida por principios claves de automatización, colaboración y mejoras continuas en todo el ciclo de vida del producto, para el despliegue exitoso de aplicaciones [1] [2].

DevOps funciona gracias a su sistema de: desarrollo, planificación, pruebas de funcionamiento, registros, configuración, lanzamiento, control y mantenimiento en las diferentes etapas de desarrollo, a través de sus herramientas que permiten la automatización de procesos.

Este sistema, también es conocido como el ciclo de vida de DevOps y se encuentra representado en la Figura 1.1.



Figura 1.1 Ciclo de vida de DevOps [3]

El conjunto de prácticas, principios y herramientas, hacen que DevOps sea una solución viable para el lanzamiento de *software* de alta calidad con tiempos de respuesta relativamente cortos. Cabe mencionar, que esta metodología realiza

una retroalimentación constante para promover su mejora continua en base a los resultados obtenidos [3].

WEB HOSTING CONTROL PANEL SOFTWARE

El panel de control es una herramienta con interfaz gráfica de usuario que permite el acceso en línea por un navegador *web*, brindando la posibilidad de administrar sitios *web*, cuentas de alojamiento *web* e incluso servidores. Cada panel de control cuenta con módulos que caracterizan su funcionamiento y determinan su calidad [4] [5]. La Figura 1.2 representa de forma general una plataforma de panel de control.



Figura 1.2 *Web hosting control panel software* [6]

Sin embargo, el *web hosting control panel software* cuentan con tareas comunes, como la gestión de dominios y archivos, configuración de cuentas, instalación de aplicaciones, administración de bases de datos, implementación de reglas de seguridad, monitorización de recursos, entre otros. Cada uno de los paneles de control tienen como finalidad aportar un servicio fácil y rápido [6] [7].

DOCKER

Docker es la tecnología que permite construir, probar y desplegar aplicaciones dentro de entornos virtuales ligeros, llamados contenedores. Gracias a su rendimiento, portabilidad y compatibilidad con los sistemas operativos, los usuarios pueden monitorear sus aplicaciones [8] [9]. *Docker* es representado por la Figura 1.3



Figura 1.3 *Docker* [10]

Docker ofrece un ambiente amigable al usuario, ya que gestiona el arranque y detención de contenedores, con la fácil implementación de su API o línea de comandos. La ventaja principal de esta herramienta es que permite la contenerización de aplicaciones, para que puedan ser desplegadas en diferentes entornos sin presentar inconvenientes [11].

Por otro lado, la eficiencia de *Docker* en comparación con las máquinas virtuales es mayor, debido a que los contenedores comparten el mismo núcleo del sistema operativo, con el propósito de garantizar la administración adecuada de recursos y el despliegue rápido de aplicaciones. Esto ha permitido relacionar de manera positiva a *Docker* con DevOps para establecer un entorno colaborativo [12].

2 METODOLOGÍA

Para el estudio de DevOps, *web hosting control panel software* y contenedores, se llevó a cabo una investigación bibliográfica, ya que la información fue recopilada y analizada de páginas web, blogs, libros y *papers*. La investigación realizada aportó de manera significativa al entendimiento general de las herramientas empleadas para el despliegue de *web hosting control panel software* en contenedores. No obstante, también fue necesario utilizar recursos audiovisuales que complementaron la investigación, con el objetivo de simplificar el uso de *Docker* y *Dockerfile*.

Al contar con la información y el dominio básico de *Docker*, se realizaron pruebas sencillas de funcionamiento, tanto en la aplicación de *Docker* en *Windows* y en máquinas virtuales con los sistemas operativos Ubuntu 22.04 y Centos 7, previamente configurados con los repositorios *Docker*. Gracias a las pruebas

efectuadas, se pudo identificar el ambiente adecuado para desarrollar el contenedor.

Para la construcción de la imagen del panel de control, se analizaron algunas instrucciones y directrices, que garantizan una buena práctica de implementación. También, se seleccionó la imagen base que va a proporcionar los recursos iniciales en la construcción de la imagen y de igual manera el *software* del panel de control a implementar, en este caso *Webmin*.

En base al sistema operativo, los paquetes que poseen los archivos y metadatos deben constar con un formato compatible, además de proporcionar todas las dependencias necesarias para que estos archivos puedan ser descomprimidos y ejecutados dentro del contenedor para instalar y administrar el *software*.

Ya establecido el panorama, se creó el archivo de texto *Dockerfile*, con Centos 7 como imagen base y las instrucciones precisas para el despliegue del software en el contenedor. En seguida se guardó la información y se implantó los comandos que permiten la construcción de la imagen con su nombre y etiqueta. Al contar con la API de *Docker*, el usuario puede crear el contenedor y establecer el nombre y puerto de acceso de manera intuitiva directamente en la plataforma, en lugar de usar el terminal de comandos.

Al crearse el contenedor, automáticamente se empiezan a generar logs que proporcionan información sobre la ejecución del sistema, además a través del terminal se pudo acceder a la línea de comandos del contenedor y configurar las credenciales de acceso a la plataforma de *Webmin*.

Para ingresar a la página de acceso del panel de control, en la barra de direcciones se fijó la dirección web `https://localhost:10000`, una vez que se haya ingresado las credenciales que fueron establecidas, se podrá visualizar la plantilla de inicio de *Webmin* con información del sistema en el que se encuentra implementado, además de permitir evidenciar el funcionamiento con las opciones de gestión en su menú.

3 RESULTADOS

En esta sección, se presenta la investigación realizada sobre las herramientas de DevOps, con el fin de comprender su funcionamiento, además se establecen parámetros que permiten desarrollar el componente práctico del presente proyecto, y se lleva a cabo la implementación del *software Webmin* con las configuraciones pertinentes para el acceso a la plataforma a través del navegador *web*.

Finalmente, se evidencia el funcionamiento de la plataforma y se brinda soluciones a inconvenientes en *Webmin*.

3.1 ANÁLISIS DE LA HERRAMIENTA DE DEVOPS

La implementación de DevOps trae consigo varios beneficios que garantizan la calidad y eficiencia del desarrollo de *software*, así como la gestión de aplicaciones. A continuación, se presentan algunos beneficios relevantes de DevOps:

- Utiliza la integración continua, como una práctica que beneficia al proceso de desarrollo de *software* ya que, automatiza la integración de varios códigos, para obtener un único *software* controlado y funcional [13].
- Consta con un sistema de entrega continua que permite automatizar tareas y aumentar la tasa de despliegues de *software* con versiones estables.
- El sistema de DevOps trabaja con herramientas técnicas, como: *Docker*, *Kubernetes*, *Ansible*, *Git*, *Jenkins*, etc., que posibilitan el intercambio de información entre los equipos de desarrollo y operación para el monitoreo y control de errores [14] [15].
- Garantiza la seguridad de aplicaciones con la implementación de *Security by Design*, enfocado al desarrollo de *software* y *hardware* para el control de amenazas y vulnerabilidades, desde el inicio del ciclo de vida del *software*.

- *DevOps* es un sistema abierto que integra nuevas herramientas conforme los avances tecnológicos se van presentando, para ofrecer una mayor eficiencia y cooperación [15].
- La utilización de *Cloud computing* ofrece el acceso a recursos de *hardware* escalables que pueden desplegar nuevas actualizaciones y ofertas de *software* con mayor facilidad y rapidez, haciendo de DevOps una herramienta escalable, eficiente y esencial para el desarrollo de aplicaciones en la nube [16] [17].

Las herramientas de DevOps son ampliamente utilizadas en el área de desarrollo y gestión empresarial, ya que brindan confiabilidad e innovación y se encuentran categorizadas en: control de versiones, integración continua CI, entrega continua CD, infraestructura, gestión de configuración, gestión de contenedores, pruebas automatizadas, monitoreo, seguridad e incidentes [18].

Para desarrollar el objetivo principal del presente proyecto, es necesario profundizar el tema de contenedores y su funcionamiento, ya que es un elemento clave para el despliegue del *software* de panel de control.

CONTENEDORES

Son paquetes que contienen los ejecutables, bibliotecas, archivos de configuración y códigos binarios necesarios para la ejecución de aplicaciones en diferentes sistemas operativos. Los contenedores a diferencia de las máquinas virtuales se caracterizan por no tener su sistema operativo, haciéndolos ligeros y portátiles. En la actualidad existen varias plataformas que garantizan su desarrollo, como: *Docker*, *Linux-vServer*, *LXC*, *LXD*, *OpenVZ*, *Systemd-nspawn*, entre otros [19].

En la Figura 3.1 se muestra la estructura de las máquinas virtuales y contenedores para comprender de mejor manera en que se diferencian.



Figura 3.1 Máquinas Virtuales vs Contenedores [20]

Los contenedores cuentan con recursos de memoria, *CPU*, redes aislados y bloqueo de entrada/salida y comparten el *kernel* del sistema en el que se encuentran. Los contenedores utilizan *namespaces* y *cgroups* para la administración de recursos y la eficiencia de los entornos aislados [20].

En la Tabla 3.1 se presentan las ventajas y desventajas de implementar contenedores.

Tabla 3.1 Ventajas y desventajas de los contenedores [20]

Ventajas	Desventajas
Presenta bajo consumo de recursos de memoria, capacidad de procesamiento y espacio en disco.	Al migrar las aplicaciones pueden verse afectadas al no ser completamente compatibles con el contenedor por dependencias específicas del sistema operativo.
Con los repositorios y repetibilidad se permite compartir contenedores en diferentes entornos de desarrollo de manera coherente y confiable.	
Presentan herramientas de monitoreo, despliegue, escalabilidad, etc.	Si existiese alguna vulnerabilidad en el <i>kernel</i> , todos los contenedores se verían afectados.
Pueden generar entornos de desarrollo y pruebas.	

DOCKER

Docker es el *software* más popular para desempeñar el desarrollo de aplicaciones en contenedores, ya que ofrece una infraestructura basada en contenedores y cuenta con los siguientes conceptos básicos [21]:

- *Images*: plantilla o archivo inmutable que contiene todos los lineamientos para ejecutar una aplicación.
- *Dockerfile*: archivo donde se definen instrucciones para crear la imagen.
- *Containers*: corresponde a la imagen creada en funcionamiento.
- *Registry*: se encuentra relacionado con *DockerHub*, ya que permite registrar las imágenes *Docker*.
- *Volumes*: son medios que permiten compartir y almacenar datos después de haber detenido el contenedor.

Docker, también cuenta con herramientas que simplifican procesos de desarrollo y despliegue de forma consistente y eficiente al empaquetar y distribuir el *software* [21].

Docker está implementado por el archivo *Dockerfile* y sus comandos de construcción y arranque. *Dockerfile* es definido como un archivo de texto simple, donde se definen instrucciones concretas para crear nuevas imágenes. Las instrucciones son los comandos claves que van a realizar acciones en la imagen base, por lo que la estructura de un archivo *Dockerfile*, comúnmente se encuentra definido de la siguiente forma [21]:

- *FROM*: permite definir la imagen base.
- *MAINTAINER*: hace referencia al autor o creador de la imagen.
- *RUN*: ejecuta comandos en la imagen base, para actualizar o realizar tareas.
- *ADD/COPY*: permite copiar o agregar archivos a la imagen desde cualquier equipo local.
- *EXPOSE*: señala el puerto de acceso al contenedor.

- *CMD*: permite ejecutar la acción del contenedor al ejecutarse por defecto.

La selección de la imagen base, la descarga e instalación de dependencias y los comandos que permiten el arranque del contenedor, representan el flujo típico de *Dockerfile* [21] [22]. La Figura 3.2 presenta un ejemplo de la estructura para la elaboración de una imagen en un archivo *Dockerfile*.

```
# Imagen base
FROM node:alpine

# Carpeta de trabajo
WORKDIR /usr/app

# Copiado de archivos
COPY ./ ./

# Instalación de dependencias
RUN npm install

# Comando por defecto del contenedor
CMD ["npm", "start"]
```

Figura 3.2 Fichero *Dockerfile* [23]

Los comentarios e instrucciones del archivo *Dockerfile* son importantes para la identificación de procesos e interpretación del código, durante la construcción de la imagen los comentarios son eliminados por encontrarse anteceditos por **#**.

Para crear la imagen se establece un proceso de construcción denominado proceso *build*, ejecutado con el comando **docker build** seguido del indicador **-t**, para asignar un nombre y etiqueta a la nueva imagen.

En base a la imagen se crea el contenedor, si la construcción se realiza en línea de comandos, se ejecuta el comando **docker run** seguido del parámetro **-p**, para definir los puertos y **-name** para indicar el nombre del contenedor y la imagen. El proceso de construcción es representado por la Figura 3.3.

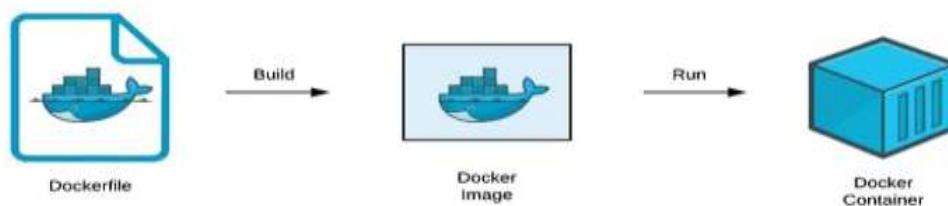


Figura 3.3 Proceso de construcción de una imagen *Docker* [24]

En caso de utilizar la aplicación de *Docker*, simplemente se corre la imagen y se llena los parámetros mostrados para definir nombre y puerto, así como la implementación de volúmenes, entre otros [25].

WEB HOSTING Y WEB HOSTING CONTROL PANEL SOFTWARE

Web hosting, también conocido como alojamiento *web*, es la herramienta que permite almacenar contenido en línea proporcionando un lugar seguro y estable, por lo que el alojamiento *web* es la parte crítica en el desarrollo de empresas virtuales y otras áreas de implementación. Es importante considerar que el *web hosting* tiene como principal ventaja reducir costos de almacenamiento y espacio físico, además de brindar servicios de seguridad y soporte [26].

Para garantizar un rendimiento próspero de los sitios *web*, es necesario considerar la capacidad y el mantenimiento adecuado del alojamiento *web*, con el fin evitar procesos de impacto negativo para la organización o privatizar al usuario de acceder a información [26].

Gracias al crecimiento exponencial de Internet y al desarrollo virtual empresarial, los *web hostings* son ampliamente utilizados a nivel mundial y como resultado de esta evolución, varias empresas se han especializado en brindar servicios de *web hosting* ajustados a las necesidades del usuario [26] [27].

WEB HOSTING CONTROL PANEL SOFTWARE

Como se mencionó anteriormente, el *software* de panel de control ofrece varias funcionalidades para la gestión y administración en servidores *web* o cuentas de *hosting*, lo que simplifica tareas como la creación de sitios *web* y la instalación de aplicaciones. Para administrar el *web hosting*, existen varias aplicaciones de panel de control que son populares, como *cPanel*, *Plesk*, *VirtualMin*, *Webmin*, entre otros.

Cada una de ellas ofrece características y funcionalidades específicas, además comparten la ventaja de facilitar la administración y gestión, brindando a los usuarios un mayor control sobre el entorno *web* [28]. La Figura 3.4 muestra las plataformas más utilizadas en la actualidad.



Figura 3.4 *Web hosting control panel software* [29]

A continuación, se presenta la descripción de funciones y características de dos paneles de control más comunes.

CPANEL

Es el primer *software* establecido para la gestión de alojamiento *web* y servidores, fue desarrollado por Jhon Nick Koston en 1996. Dado a su continuo desarrollo desde su despliegue, *cPanel* cuenta con una funcionalidad extra, denominada *WHM* o *Web Hosting Manager* y está enfocado en los sistemas operativos *Centos* y *Red Hat*.

La diferencia radica en que *cPanel* corresponde a la interfaz de monitoreo de alojamiento *web* para los clientes, mientras que *WHM* es la interfaz utilizada por los proveedores para la administración del servidor *web*. *cPanel* es un *software* completo que ofrece servicios íntegros y estables, sin embargo, no es una interfaz liviana y tampoco es un *software* gratuito, por lo que es necesario pagar una licencia [29] [30].

Las funciones principales de *cPanel* se encuentran en las herramientas de publicación de sitios *web*, gestión de correos electrónicos y calendarios, copia de seguridad en la transferencia y gestión de archivos, administración de dominios, diseño y despliegue de base de datos, siendo así un entorno que brinda autonomía de gestión y facilidad de operación para usuarios que trabajan con varios sitios *web*.

En la Figura 3.5 se ilustra de mejor manera las herramientas básicas de *cPanel*.

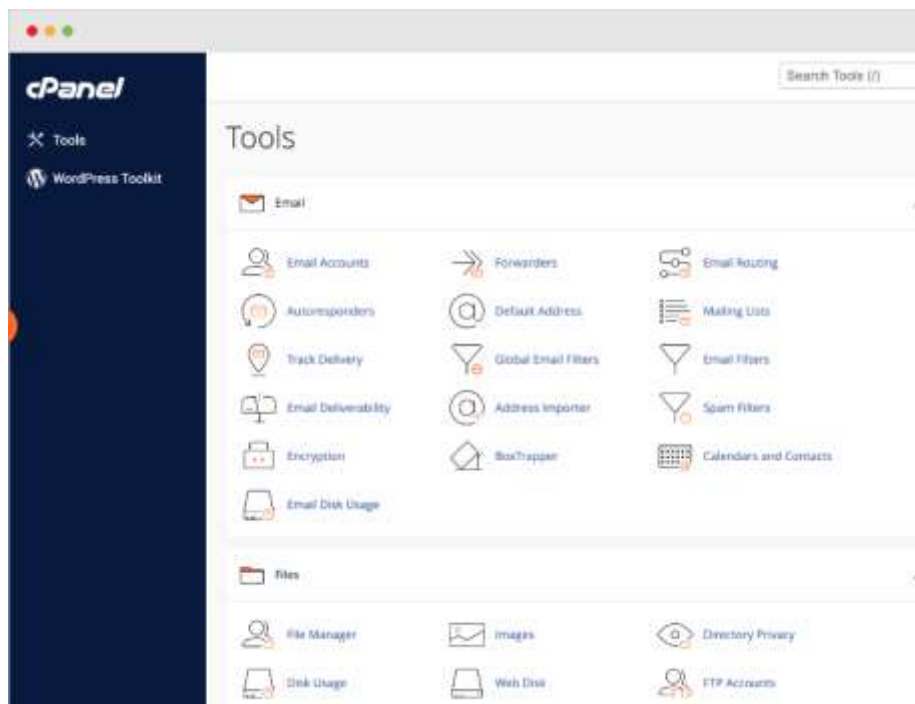


Figura 3.5 Herramientas de cPanel [31]

WEBMIN

Webmin es una plataforma de código abierto que administra servidores, está conformada por *Usermin*, para proporcionar al usuario el control administrativo de tareas y *Virtualmin* dirigido a la administración de dominios, sitios y servicios *web*, así como los recursos a nivel de proveedor [29].

Webmin a diferencia de *cPanel* es una plataforma gratuita y puede ser implementada en diferentes sistemas *Linux*. *Webmin* emplea diferentes herramientas que permiten realizar configuraciones en cuentas de usuarios y servidores como *MySQL*, *Apache*, *DNS*, *PHP*. *Webmin* usa módulos *Perl* y se comunica a través del navegador con la biblioteca *OpenSSL* por el puerto 10000 [32].

Las funciones que *Webmin* posibilita son las siguientes:

- Crear, eliminar y editar cuentas de usuario.
- Utiliza el protocolo *NFS* (sistema de archivos de red) para compartir archivos y directorios en sistemas *Linux*.
- Permite monitorear, instalar y eliminar paquetes.

- Realiza cambios de direccionamiento, *DNS* y enrutamiento.
- Proporciona mayor seguridad con configuraciones de *Firewall*.
- Administra base de datos *MySQL/PostgreSQL*.
- Usa *Samba* para el intercambio de archivos *Windows* con *Linux*.

Los repositorios de *Webmin* contienen dependencias y versiones actualizadas de la plataforma, además de brindar actualizaciones automáticas del sistema en cuestión [32]. La Figura 3.6 muestra las herramientas de configuración de la plataforma *Webmin*.



Figura 3.6 Herramientas de Webmin [32]

La combinación de *Webmin* con *Virtualmin*, hacen que la plataforma sea flexible y con mayor capacidad de administración. Los usuarios *Virtualmin* corresponde a los usuarios *Webmin*, por lo que la capacidad de delegar tareas permita a los usuarios de menor experiencia mantener servicios de correo, base de datos, *DNS* y servidores.

En resumen, *Webmin* en conjunto con *Virtualmin* ofrecen herramientas de personalización con arquitectura modular para controlar programas y crear plantillas de servidores en base a necesidades específicas, además de brindar flexibilidad de implementación y administración [33].

3.2 DISEÑAR LA SOLUCIÓN PARA CADA SERVICIO DE NETWORKING MEDIANTE HERRAMIENTAS DE DEVOPS

Teniendo en cuenta los conceptos de DevOps, contenedores, *web hosting* y *web hosting control panel software*, se procede a efectuar los recursos necesarios para la implementación de servicio de *networking*.

En primera instancia, se definen las herramientas a utilizar para el despliegue del *web hosting control panel software* en contenedores, y se selecciona el sistema operativo en el cual se va a basar la imagen a crear. Dado los antecedentes de las diferentes plataformas de panel de control, es necesario establecer un *software* libre y que sus versiones no dificulten la instalación.

Considerando lo expuesto, *Webmin* aparte de ser un *software* libre, la arquitectura modular en la que se encuentra basado le permite la adaptación y compatibilidad en los diferentes sistemas operativos *Linux*.

INSTALACIÓN DE SOFTWARES PARA LA EJECUCIÓN DE WEBMIN EN CONTENEDORES

Para crear el contenedor, se puede implementar varias tecnologías orientadas a la gestión de contenedores. En este caso, *Docker* proporciona un ambiente amigable para el despliegue de imágenes y contenedores en sistemas operativos *Linux* y *Windows*.

Docker Desktop en *Windows*, ofrece una forma visual de gestionar y administrar contenedores, por lo que, se consideró como la mejor opción para el desarrollo práctico. La instalación de *Docker Desktop* es relativamente sencilla, y se puede sincronizar con *Docker Hub*. Con la instalación completa, se obtiene la interfaz gráfica mostrada en la Figura 3.7.

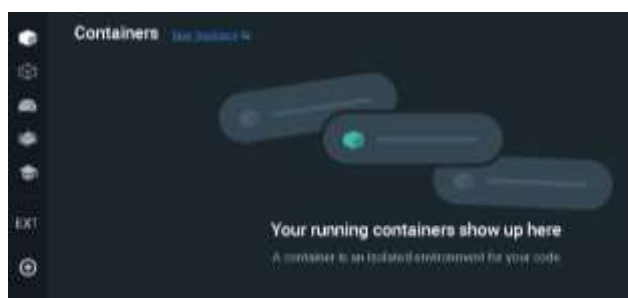


Figura 3.7 Interfaz gráfica de *Docker Desktop*

Una imagen Docker es creada en base a instrucciones que señalan la construcción y configuración de aplicaciones. Para ello, se requiere crear un archivo *Dockerfile* en el sistema operativo.

Ubuntu on Windows permite utilizar herramientas de *Linux* en un entorno *Windows*, lo que genera una compatibilidad nativa con *Docker*. Además, permite realizar pruebas de funcionamiento en aplicaciones de distribución *Linux*, como es el caso de *Webmin*.

En la instalación de *Ubuntu on Windows* se debe considerar la implementación de *WSL 2* para ejecutar contenedores en entornos compatibles con el sistema *Windows*. Finalizada la instalación, al ejecutar el comando **docker info**, se debería visualizar la información presentada en la Figura 3.8.

```
proyecto@MSI:~$ docker info
Client: Docker Engine - Community
Version: 24.0.5
Context: default
```

Figura 3.8 Información de Docker

HERRAMIENTAS Y DEPENDENCIAS PARA LA IMPLEMENTACIÓN DE WEBMIN

Como se había mencionado anteriormente, *Webmin* es compatible con varias distribuciones *Linux*, por lo que, la imagen base a implementar puede ser seleccionada según necesidades específicas, dado que es la encargada de aportar con el sistema operativo y recursos mínimos de implementación.

Considerando que *Webmin* es una plataforma de entorno empresarial, se estable el sistema operativo *CentOS* como medio de implementación del *software*, ya que se caracteriza por contar con repositorios estándar que contribuyen a la instalación de paquetes y dependencias ajustadas a las necesidades del panel de control.

En el despliegue de la imagen de *Webmin*, es esencial la implementación de módulos *Perl*. Estos módulos facultan a los usuarios realizar configuraciones y supervisiones de manera remota. Además, de proporcionar a *Webmin* una

interfaz *web* amigable, también contribuyen en tareas de administración, gestión de acceso, adaptabilidad y automatización [34].

Por otro lado, la configuración y administración de aplicaciones se realiza por paquetes *RPM (Red Hat Package Manger)* compatibles con *CentOS*, ya que cuenta con archivos y metadatos necesarios para la instalación. Los paquetes incluyen información relevante sobre la aplicación, como nombre, versión, archivos, dependencias, etc. La mejor manera de gestionar los paquetes es con herramientas de administración, como **yum** o **dnf**, ya que simplifica su administración [35].

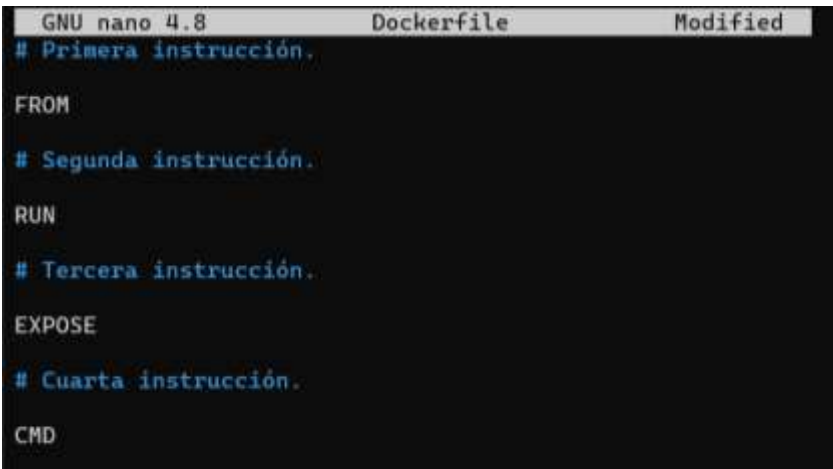
En base a lo expuesto se procede a desarrollar el *web hosting control panel software* en el contenedor.

3.3 IMPLEMENTAR LAS SOLUCIONES MEDIANTE HERRAMIENTAS DE DEVOPS PARA EL DESPLIEGUE DE LOS SERVICIOS DE *NETWORKING*

Con el ambiente *Docker* ya establecido y los recursos de instalación analizados, se desarrolla el código para la construcción de la imagen *Webmin*.

ESTRUCTURA DE LA IMAGEN *WEBMIN*

En el terminal de *Ubuntu*, se ejecuta el comando **nano Dockerfile** para crear y editar el archivo. Dentro del archivo *Dockerfile*, se van a implementar las instrucciones expuestas en la Figura 3.9.



```
GNU nano 4.8 Dockerfile Modified
# Primera instrucción.
FROM
# Segunda instrucción.
RUN
# Tercera instrucción.
EXPOSE
# Cuarta instrucción.
CMD
```

Figura 3.9 Archivo *Dockerfile*

La primera instrucción, se utiliza para especificar la imagen base a partir de la cual se va a construir la nueva imagen personalizada. La instrucción antes mencionada se muestra en la Figura 3.10.

```
FROM centos:7
```

Figura 3.10 Implementación de la Imagen base

En este contexto, se ha seleccionada la imagen **centos:7**, por su entorno fundamentado en el sistema operativo *CentOS*. Para que la imagen sea operativa es necesario emplear la segunda instrucción, ya que permite ejecutar pautas durante el proceso de construcción de la imagen y realizar tareas que contribuyen a la preparación adecuada de la imagen.

La primera acción en implementar consiste en la actualización de paquetes disponibles en los repositorios con el comando **update**, así como la instalación de herramientas que permiten descargar archivos desde internet en sistemas *Linux* y descomprimirlos de ser necesario. La Figura 3.11 muestra los comandos utilizados para la obtención y descompresión de archivos.

```
RUN yum update -y && yum install -y \  
# Herramienta de descarga  
    wget \  
# Herramienta de descompresión de archivos  
    unzip
```

Figura 3.11 Herramientas para la gestión de archivos

Dentro de la segunda instrucción, también se implementan los módulos de medida de seguridad y *Perl*, los cuales facilitan la modificación de archivos para crear imágenes según necesidades específicas y posibilitar la comunicación entre el servidor de usuario y el servidor, donde se encuentra la instancia de *Webmin* con los protocolos *SSL* y *TLS*. Los módulos se encuentran descritos en la Figura 3.12.

```

# Gestión de claves y certificados digitales
  openssl \

# Módulos Perl
# Automatización de tareas
  perl \
  perl-IO-Tty \

# Establece la comunicación entre el usuario y webmin
  perl-Net-SSLeay \

```

Figura 3.12 Módulos de automatización y seguridad

No obstante, también se requiere utilizar los módulos presentados en la Figura 3.13, para garantizar la autenticidad e integridad de los archivos que componen la imagen, además de proporcionar bibliotecas de autenticación más avanzadas y específicas.

```

# Bibliotecas de autenticación
  perl-Authen-PAM \

# Integridad y autenticidad de archivos y componentes
  perl-Digest-MD5 \
  perl-Digest-SHA \

```

Figura 3.13 Módulos de autenticación

Finalmente, se considera módulos que garanticen la depuración, construcción y operación de *scripts Perl* con la detección de codificaciones adecuadas para la interpretación precisa de datos como se indica en la Figura 3.14.

```

# Interpretación de datos
  perl-Encode-Detect \

# Construcción y depuración de paquetes con scripts Perl
  perl-Data-Dumper

```

Figura 3.14 Módulos de gestión de *scripts*

Para la descarga de *Webmin* en archivos *.rpm*, existen plataformas como *GitHub* y *SourceForge* que ofrecen paquetes actuales, auténticos y confiables. Para proceder a la descarga del paquete en *Dockerfile* se implementa la instrucción de ejecución de procesos en conjunto con la herramienta y la dirección web del archivo establecida en la Figura 3.15. De igual manera, es necesario establecer parámetros que actualizan y proporcionan información del proceso de instalación del paquete.

```
# Descarga e instalación de Webmin
RUN wget http://www.webmin.com/download/rpm/webmin-current.rpm
RUN rpm -Uvh webmin-current.rpm
```

Figura 3.15 Descarga e instalación del paquete *RPM*

Por último, se indica el puerto y el proceso para iniciar el servidor de *Webmin* con las instrucciones definidas en la Figura 3.16. Además, se llevará un control de *logs* en tiempo real, ya que proporcionan información sobre el funcionamiento y actividad del servidor web de *Webmin*.

```
# Tercera instrucción.
# Especificación de puerto Webmin
EXPOSE 10000
# Cuarta instrucción.
# Inicio del servidor web Webmin y monitoreo del mismo
CMD /etc/webmin/start && tail -F /var/webmin/miniserv.log
```

Figura 3.16 Asignación de puerto e inicio del servidor *web*

Realizados los cambios en el editor de texto *Dockerfile*, se verifica y guarda la información proporcionada. Para la construcción de la imagen e identificación, se emplea el comando de la Figura 3.17.

```
root@MSI:/home/proyecto# docker build -t webmin .
[+] Building 309.6s (5/7) docker:default
=> [internal] load .dockerignore 0.1s
=> => transferring context: 2B 0.0s
=> [internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 1.89kB 0.0s
=> WARN: Empty continuation line found in: RUN yum updat 0.1s
=> [internal] load metadata for docker.io/library/centos 1.9s
=> [1/4] FROM docker.io/library/centos:7@sha256:be65f48 14.4s
=> => resolve docker.io/library/centos:7@sha256:be65f488 0.0s
=> => sha256:be65f488b7764ad3638f236b7b5 1.26kB / 1.26kB 0.0s
=> => sha256:dead07b4d8ed7e29e98de6f4504d87e 529B / 529B 0.0s
=> => sha256:e6b8ea3f44bd8b5183bb561b4c1 2.75kB / 2.75kB 0.0s
=> => sha256:2d473b07cdd5f0912cd6f1a78 76.10MB / 76.10MB 9.7s
=> => extracting sha256:2d473b07cdd5f0912cd6f1a783352c82 4.8s
=> [2/4] RUN yum update -y && yum install -y wget unz 54.4s
=> [3/4] RUN wget http://www.webmin.com/download/rpm/w 238.7s
=> => # ..... 67% 76.5K 1m53s
=> => # 26800K .....
=> => # ..... 67% 76.5K 1m52s
=> => # 26850K .....
=> => # ..... 67% 76.8K 1m52s
=> => # 26900K .....
```

Figura 3.17 Construcción de la imagen *Webmin*

Cuando la imagen ya se haya construido, *Docker Desktop* muestra el estado de la imagen junto con su información básica como se indica en la Figura 3.18.

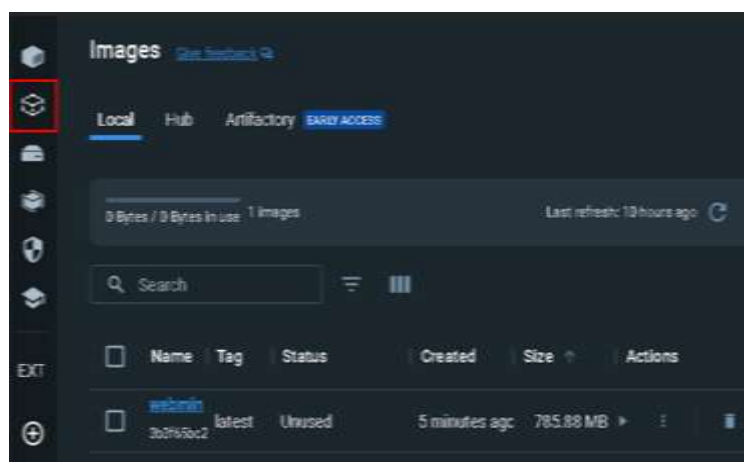


Figura 3.18 Imagen *Webmin* en *Docker Desktop*

Con *Docker Desktop* la creación de contenedores es más sencilla, ya que su interfaz gráfica proporciona plantillas y configuraciones predefinidas, por lo que la interfaz de creación permite configurar puertos, volúmenes, entornos, etc.

Los pasos para la creación del contenedor son los siguientes:

1. La Figura 3.19 muestra el inicio del contenedor con la acción **Run**.

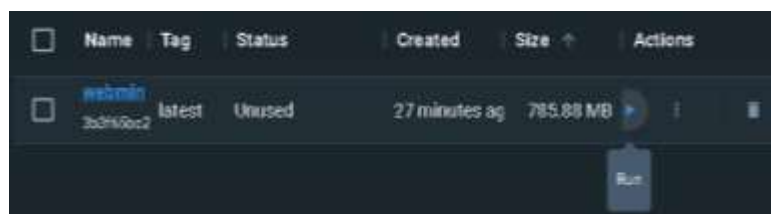


Figura 3.19 Inicio del contenedor

2. Ingresar el puerto de host y el nombre del contenedor para su adecuado funcionamiento en las secciones mostradas en la Figura 3.20.

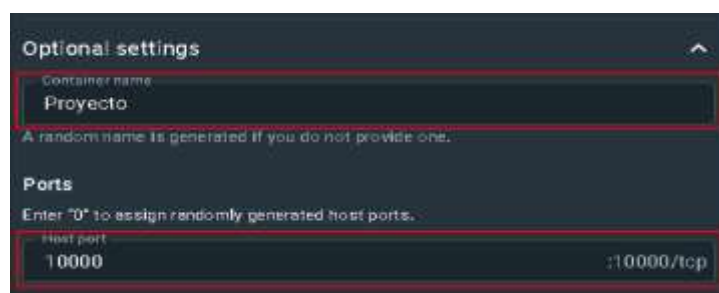


Figura 3.20 Configuración del contenedor

3. Ejecutar el contenedor.

La Figura 3.21 muestra que el contenedor se encuentra encendido y funcionando.



Figura 3.21 Ejecución del contenedor

Al iniciar el contenedor, ya se puede acceder a la aplicación mediante el navegador web.

3.4 VERIFICACIÓN DEL FUNCIONAMIENTO DE CADA SERVICIO DE *NETWORKING* IMPLEMENTADO MEDIANTE DEVOPS

En la barra de direcciones del navegador web se ingresa la dirección con la cual se va a acceder a la plataforma *Webmin* con <https://localhost:10000> y se realizar las configuraciones solicitadas para acceder a la plantilla de inicio sesión de *Webmin*.

En la Figura 3.22 se ve la página de configuración y en la Figura 3.23 la plantilla de acceso.



Figura 3.22 Configuración de acceso

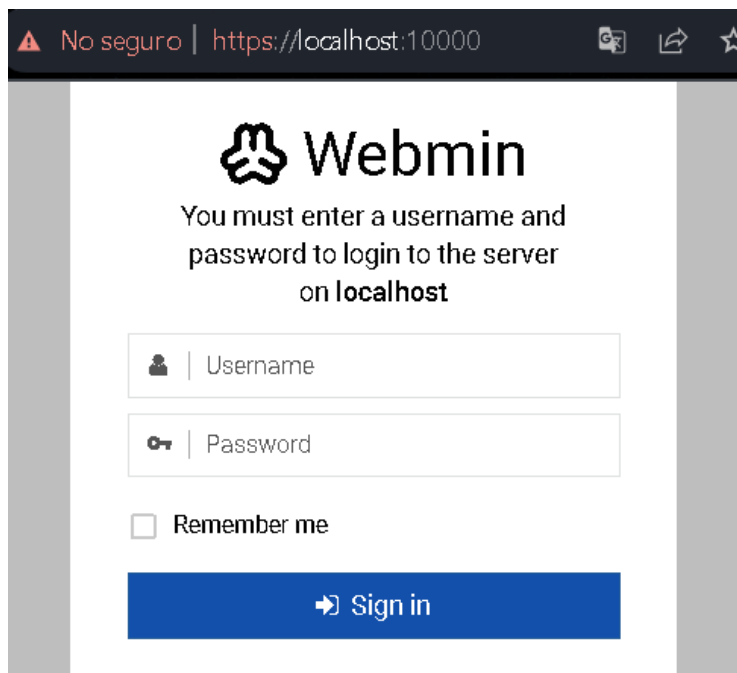


Figura 3.23 Inicio de sesión de *Webmin*

Sin la configuración de las credenciales de acceso no se puede ingresar a la plataforma *Webmin*, por ende, se debe implementar el comando señalado en la Figura 3.24, para ingresar al contenedor, ya que toda configuración relacionada a la aplicación debe realizarse dentro del contenedor.

```
root@MSI:/home/proyecto# docker exec -it Proyecto /bin/bash
[root@4b37f700732d /]#
```

Figura 3.24 Acceso al terminal del contenedor

Dentro del terminal del contenedor se ejecuta el comando **passwd** seguido del usuario para otorgar una nueva contraseña y se reinicia el contenedor.

```
[root@4b37f700732d /]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@4b37f700732d /]#
```

Figura 3.25 Nueva contraseña de usuario *root*

Sin embargo, *Webmin* no va a poder desenvolverse de manera adecuada si no se han implementado las herramientas de configuración e información relacionadas a la red e interfaces de red. Con el fin de solventar este problema, se instalan las herramientas básicas en el contenedor con el comando de la Figura 3.26.

```
[root@4b37f700732d /]# yum install -y net-tools
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
 * base: mirror.cedia.org.ec
 * extras: mirror.cedia.org.ec
 * updates: mirror.cedia.org.ec
Resolving Dependencies
--> Running transaction check
--> Package net-tools.x86_64 0:2.0-0.25.20131004git.el7 will be
installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package      Arch      Version                               Repository
Size
=====
Installing:
net-tools    x86_64    2.0-0.25.20131004git.el7             base      386 k
```

Figura 3.26 Instalación de herramientas para *ifconfig* e *IP*

Ya realizados los cambios, se puede acceder a la plataforma con las credenciales configuradas. La plantilla de inicio de *Webmin* brinda información el estado del sistema en donde se implementó el panel de control. Al igual que la interpretación estadística de los recursos de *CPU*, memoria real / virtual y espacio en el disco local.

La información es proporcionada en la Figura 3.27 y la Figura 3.28.



Figura 3.27 Información del sistema



Figura 3.28 Estadísticas del estado del sistema

Webmin cuenta con un amplio menú de herramientas como se muestra en la Figura 3.29 que permiten realizar cambios personalizados en la plataforma, como: cambio de idioma, modo de presentación, cambiar puerto de acceso, copia de seguridad de archivos y configuración, cambio de contraseñas, gestión de usuarios y grupos, etc.

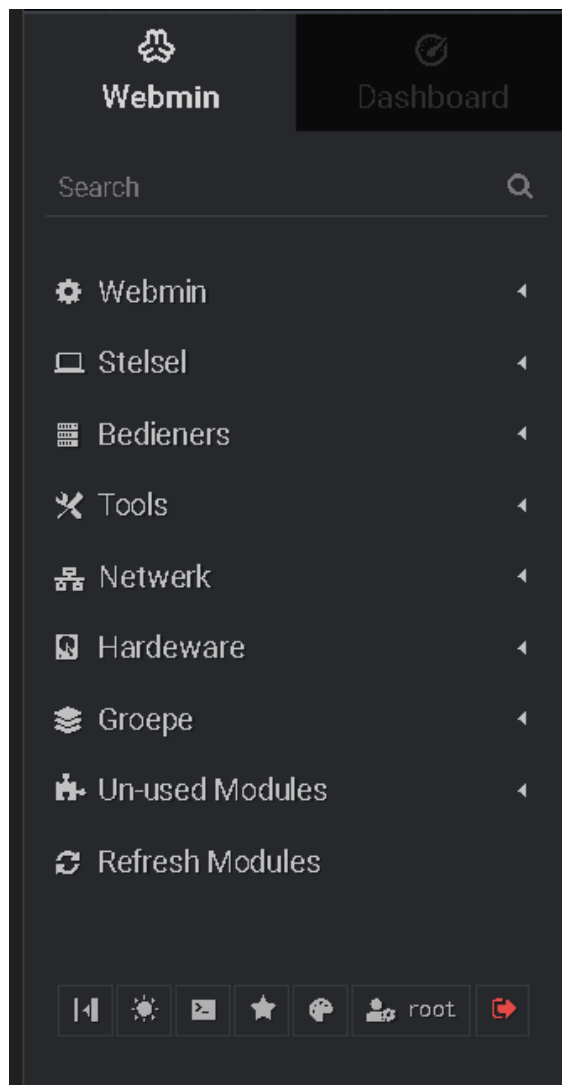


Figura 3.29 Menú principal

Para personalizar la plataforma, se van a realizar cambios básicos que permiten entender de mejor forma el funcionamiento de la plataforma.

Lo primero a realizar, corresponde al cambio de idioma. En el menú principal, ingresar a *Webmin* y seleccionar la opción que permite realizar los cambios relacionados a idioma y tema, la Figura 3.30 detalla los pasos a seguir.

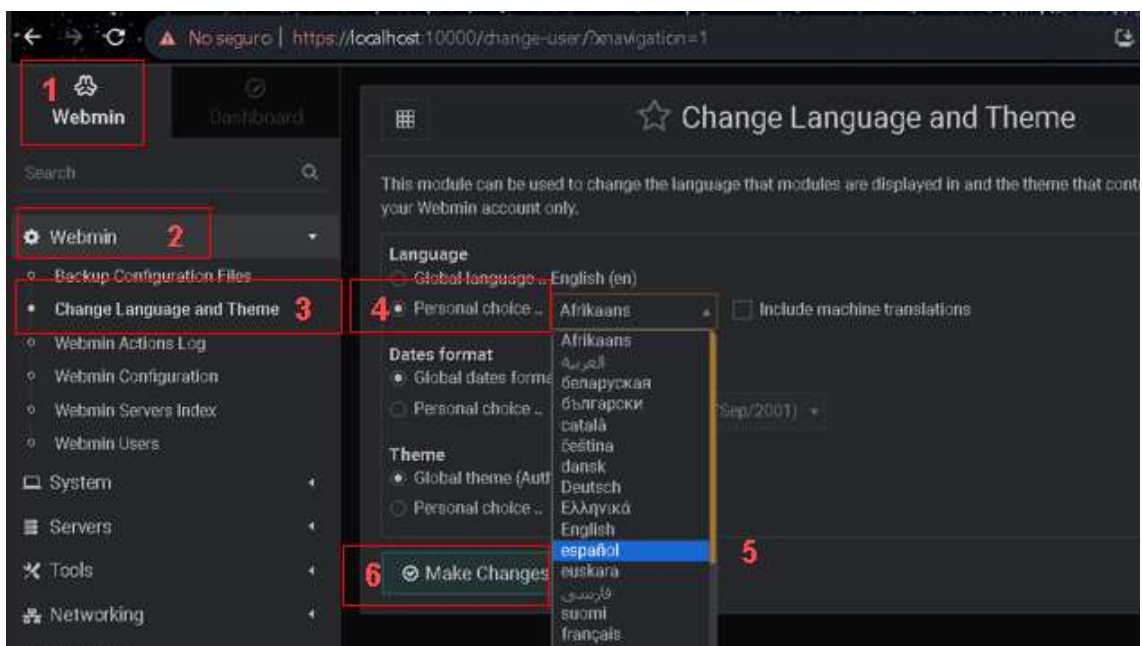


Figura 3.30 Configuración de idioma

En un ambiente laborar se puede establecer usuarios con acceso a diferentes herramientas, haciendo de *Webmin* un ambiente definido para ciertas áreas de monitoreo y configuración. A continuación, en la Figura 3.31 y Figura 3.32 se implementa la herramienta de **Usuarios Webmin** para establecer un tipo de ambiente limitado.

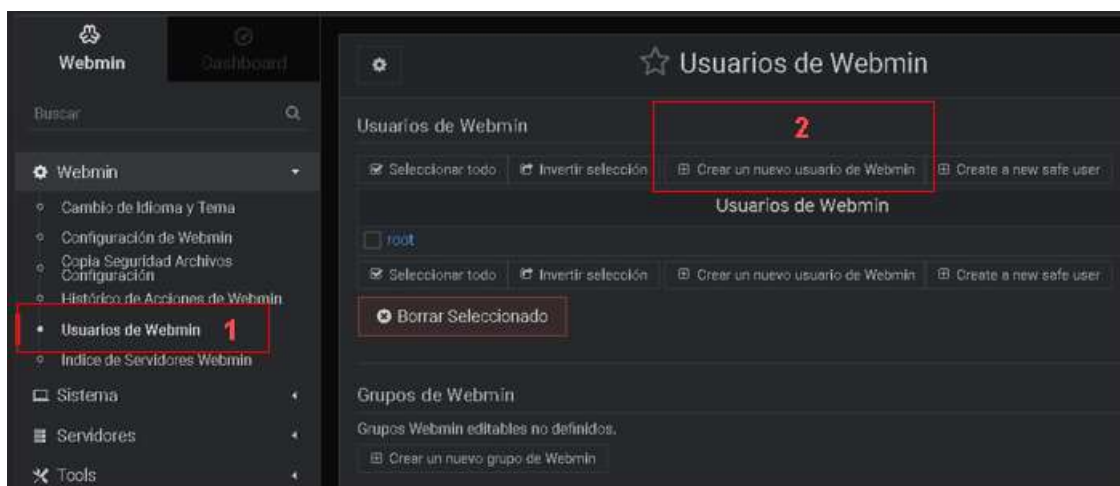


Figura 3.31 Nuevo usuario *Webmin*

← [Icono] ☆ Crear Usuario de Webmin

▼ Derechos de acceso de usuarios Webmin

Nombre de usuario
adriana

Contraseña
Configurar a ... adriana123 Forzar cambio en el siguiente ingreso

Nombre real
usuario1

Privilege level
Unrestricted

► Opciones de interfaz de usuario

► Opciones de seguridad y limites

► Modulos de webmin disponibles

⊕ Crear

Figura 3.32 Configuración de credenciales

En la sección de módulos disponibles, se va a limitar el acceso a herramientas de administración y gestión, como se indica en la Figura 3.33.

▼ Modulos de webmin disponibles

Seleccionar todo Invertir selección

Webmin

Cambio de Idioma y Tema
 Configuración de Webmin
 Historico de Acciones de Webmin
 Usuarios de Webmin

Configuración de Usermin
 Copia Seguridad Archivos Configuración
 Scheduled Webmin Functions
 Índice de Servidores Webmin

Sistema

Autenticación PAM
 Bootup and Shutdown
 Comandos Planificados
 Cuotas de Disco
 LDAP Client
 Paquetes de Software
 Páginas del Manual
 Sistemas de Archivo de Disco y Red
 System Logs NG
 System Status
 Usuarios y Grupos

Bacula Backup System
 Cambio de Contraseñas
 Copia de Seguridad de Sistema de Archivos
 Históricos (Logs) del Sistema
 MIME Type Programs
 Procesos en curso
 Rotación de Históricos (Logs)
 Software Package Updates
 System Logs Viewer
 Tareas Planificadas (Cron)
 Usuarios y Grupos LDAP

Servidores

Compartición de Archivos de Windows mediante Samba
 Configuración de QMail

Dovecot: Servidor de IMAP/POP3
 Fetchmail - Descarga de correo
 Generador de Informes de Análisis de Squid
 Lectura de Correo de Usuarios
 Servidor ProFTPD
 Servidor Web Apache
 Servidor de Base de Datos PostgreSQL
 Servidor de DNS BIND
 Squid - Servidor Proxy

Configuración de Postfix
 Configuración de Sendmail
 Edim Mailserver
 Filtro de Correo Procmail
 LDAP Server
 OpenSLP Server
 Servidor SSH
 Servidor de Base de Datos MySQL
 Servidor de DHCP
 SpamAssassin - Filtro de Correo
 Webalizer - Análisis de Históricos (Logs)

Figura 3.33 Selección de herramientas para el nuevo usuario

Cuando se tengan guardadas los cambios y generado el nuevo usuario, se podrá acceder a la plataforma con las credenciales establecidas para ese usuario, esta plataforma contará solo con las herramientas señaladas anteriormente, haciendo a *Webmin* un ambiente limitado para el nuevo usuario, tal como se muestra en la Figura 3.34.

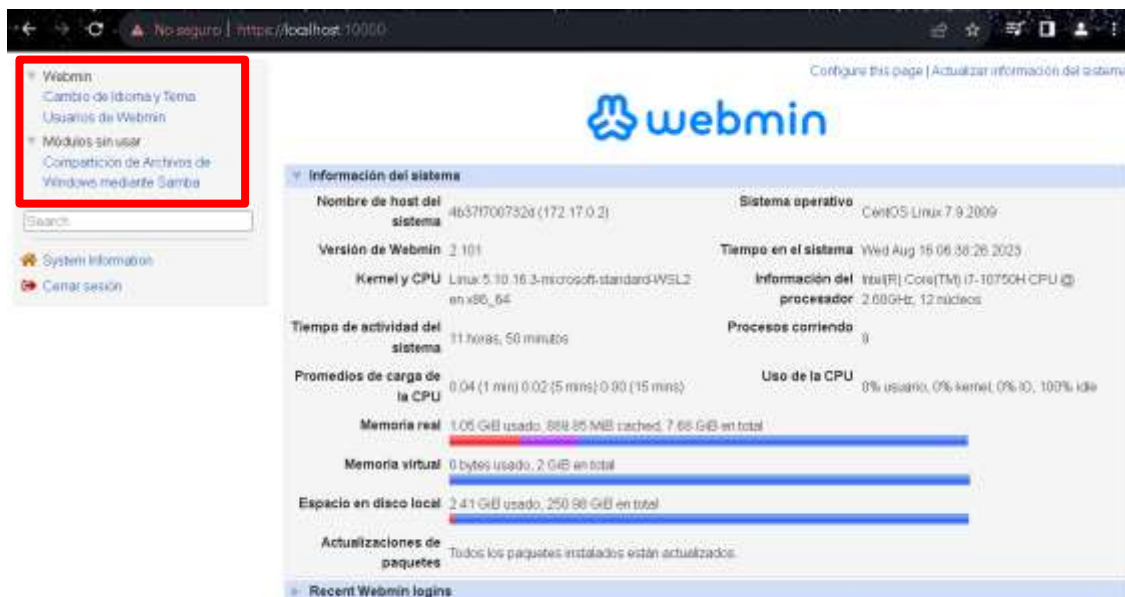


Figura 3.34 Plataforma de *Webmin* para el nuevo usuario

Las herramientas de *Webmin*, hacen de la plataforma un medio entendible y funcional en los diversos ambientes de aplicación. Por lo tanto, la personalización del sistema está directamente relacionado a las necesidades del usuario y de su ambiente de aplicación.

4 CONCLUSIONES

- DevOps engloba un conjunto de culturas y metodologías durante el proceso de desarrollo, implementación y mantenimiento de un *software*, enfocados a la automatización y colaboración entre los equipos de desarrollo y *TI* con herramientas de innovación para el despliegue de nuevos productos.
- Los servicios de DevOps tienen como finalidad ofrecer eficiencia, velocidad y seguridad en ambientes *networking*, por lo que tienen la capacidad de adaptarse a las demandas y necesidades de la organización, además de gestionar y presentar una red confiable y robusta.
- *Docker* y *Docker Desktop* permiten gestionar aplicaciones, con el fin de garantizar su agilidad, confiabilidad, portabilidad, aislamiento, eficiencia y escalabilidad mediante pruebas de funcionamiento y control de cambios para entregar un *software* de alta calidad con versiones adaptadas a nuevas necesidades.
- *Web hosting control panel software* corresponde a un grupo de herramientas que simplifica la administración, gestión y configuración de servicios de *networking* relacionados al alojamiento *web*.
- *cPanel* y *Webmin* son plataformas que permiten gestionar los recursos y servicios de alojamiento *web*. Cada una cuenta funcionalidades específicas que las hacen viables según el área de aplicación.
- *Webmin* cuenta con archivos *Perl* utilizados para implementar funcionalidades y características en la plataforma que benefician a la adaptabilidad del software en diferentes áreas de desarrollo empresarial.
- La construcción de una imagen *Docker* se encuentra basada en la implementación de una imagen ya creada, a dicha imagen se la denominada imagen base. La estructura de una imagen *Docker* es diseñada en base a las necesidades y requisitos de la aplicación a desplegarse en el contenedor.
- Los archivos que contienen aplicaciones, dependencias y metadatos en distribuciones *Linux* pueden ser identificados por su tipo de extensión, es

decir que los archivos compatibles a *CentOS*, *Fedora* y *Red Hat Enterprise Linux* tienen extensión **.rpm**. En el caso de *Debian* y *Ubuntu* se implementan los archivos **.deb**. La implementación de los archivos **.deb** o **.rpm**, dependerá directamente del sistema operativo en el cual se desarrolla la imagen *Docker*.

5 RECOMENDACIONES

- Para la elaboración de una imagen *Docker*, se debe considerar su estructura y configuración, ya que son elementos claves para garantizar su rendimiento, mantenimiento, eficiencia, tamaño, etc.
- La imagen base en un archivo *Dockerfile*, debe ser seleccionada en base a su eficiencia, estabilidad y compatibilidad con la aplicación.
- Para la gestión de los archivos *RPM*, es imprescindible garantizar la autenticidad de los *scripts* contenidos.
- Los cambios efectuados en el archivo *Dockerfile* deben ser verificados y almacenados para la construcción exitosa de la imagen.
- Al crear el contenedor se debe señalar el puerto de acceso a la aplicación, por lo que en el caso de *Webmin* el puerto del contenedor y el puerto de acceso a plataforma es **10000**.
- Es recomendable definir credenciales de acceso a la plataforma para mejorar la seguridad e integridad del sistema.
- *Webmin* gestiona y analiza la red, por lo que es necesario la implementación de herramientas de red dentro del contenedor.
- La personalización de *Webmin*, debe realizarse en base a las necesidades y vulnerabilidades de su entorno.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] «NetApp,» [En línea]. Available: <https://www.netapp.com/es/devops-solutions/what-is-devops/>. [Último acceso: 12 Junio 2023].
- [2] «Atlassian,» [En línea]. Available: <https://www.atlassian.com/es/devops>. [Último acceso: 12 Junio 2023].
- [3] A. Pathak, «GEEKFLRE,» DevOps, 20 Diciembre 2022. [En línea]. Available: <https://geekflare.com/es/devops-lifecycle/>. [Último acceso: 14 Junio 2023].
- [4] K. Horne, «Digital,» 30 Marzo 2023. [En línea]. Available: <https://digital.com/best-web-hosting/control-panels/>. [Último acceso: 13 Junio 2023].
- [5] «How To Hosting,» [En línea]. Available: <https://hth.guide/es/definiciones/what-is-web-hosting-control-panel/>. [Último acceso: 14 Junio 2023].
- [6] A. B., «Hostinger Tutoriales,» 8 Junio 2023. [En línea]. Available: <https://www.hostinger.com/tutorials/hosting-control-panels/>. [Último acceso: 14 Junio 2023].
- [7] J. C., «Web Hostface,» 23 Marzo 2021. [En línea]. Available: <https://www.webhostface.com/blog/choosing-web-hosting-control-panel/>. [Último acceso: 14 Junio 2023].
- [8] «Kinsta,» 13 Enero 2023. [En línea]. Available: <https://kinsta.com/es/base-de-conocimiento/que-es-docker/>. [Último acceso: 14 Junio 2023].
- [9] C. D., «Hostinger,» 4 Junio 2023. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-docker>. [Último acceso: 14 Junio 2023].

- [10] J. B. Scott McCarty, «Red Hat Developer,» Red Hat, 15 Mayo 2014. [En línea]. Available: <https://developers.redhat.com/blog/2014/05/15/practical-introduction-to-docker-containers>. [Último acceso: 15 Junio 2023].
- [11] «UNIR,» 11 Mayo 2023. [En línea]. Available: <https://mexico.unir.net/ingenieria/noticias/que-es-docker/#:~:text=Docker%20permite%20que%20tanto%20una,sea%20m%C3%A1s%20f%C3%A1cil%20y%20eficiente..> [Último acceso: 15 Junio 2023].
- [12] A. E. Calle, «Autentia,» Autentia, 25 Febrero 2020. [En línea]. Available: <https://www.adictosaltrabajo.com/2020/02/25/dockeriza-tu-api-explicado-para-principiantes/#:~:text=Docker%20es%20un%20proyecto%20de,aplicaciones%20en%20m%C3%BAltiples%20sistemas%20operativos..> [Último acceso: 15 Junio 2023].
- [13] V. Cuervo, «AiT,» 23 Marzo 2019. [En línea]. Available: <https://www.arquitectoit.com/devops/7-beneficios-devops/>. [Último acceso: 18 Junio 2023].
- [14] A. d. I. Peña, «Hiberus,» 27 Marzo 2023. [En línea]. Available: <https://www.hiberus.com/crecemos-contigo/herramientas-devops-para-mejorar-la-eficiencia-y-la-colaboracion/#:~:text=Los%20equipos%20de%20DevOps%20trabajan%20juntos%20para%20garantizar%20la%20entrega,Ansible%2C%20Nagios%20y%20ELK%20Stack..> [Último acceso: 23 Junio 2023].
- [15] N. Montaña, «AMBIT BST,» 26 Noviembre 2019. [En línea]. Available: <https://www.ambit-bst.com/blog/10-ventajas-de-implementar-devops>. [Último acceso: 19 Junio 2023].
- [16] «Intel,» 24 Enero 2022. [En línea]. Available: <https://www.intel.com/content/www/us/en/cloud-computing/devops.html#:~:text=DevOps%20is%20a%20software%20engi>

neering,and%20services%20in%20the%20cloud.. [Último acceso: 23 Junio 2023].

- [17] F. d. J. N. C. Ana María Felipe Redondo, DevOps: un vistazo rápido, DevOps: a quick look, Estado de Hidalgo: 05/07/2021, 2021.
- [18] E. P. Saguier, «Invgate,» 10 Octubre 2022. [En línea]. Available: <https://blog.invgate.com/es/herramientas-devops>. [Último acceso: 25 Junio 2023].
- [19] «Netapp,» [En línea]. Available: <https://www.netapp.com/es/devops-solutions/what-are-containers/>. [Último acceso: 3 Julio 2023].
- [20] A. R. & A. Som, «Ebury,» Mayo 2022. [En línea]. Available: <https://osl.ugr.es/wp-content/uploads/2022/05/Contenedores-Iniciacio%CC%81n-a-Docker-y-casos-de-uso-pra%CC%81cticos-Mayo2022.pptx.pdf>. [Último acceso: 3 Junio 2023].
- [21] J. Gomez, «YouTube,» JGAITPro, 22 Marzo 2017. [En línea]. Available: <https://www.youtube.com/watch?v=mPB2W1255iE>. [Último acceso: 3 Julio 2023].
- [22] «Microsft,» 20 Marzo 2023. [En línea]. Available: <https://www.ramoncarrasco.es/es/content/es/kb/150/estructura-del-fichero-dockerfile>. [Último acceso: 1 Julio 2023].
- [23] R. Carrasco, «ramoncarrasco.es,» [En línea]. Available: <https://www.ramoncarrasco.es/es/content/es/kb/150/estructura-del-fichero-dockerfile>. [Último acceso: 2 Julio 2023].
- [24] R. Chen, «Medium.com,» 2 Diciembre 2020. [En línea]. Available: <https://medium.com/swlh/understand-dockerfile-dd11746ed183>. [Último acceso: 5 Julio 2023].
- [25] S. Afreen, «Simpli Learn,» 17 Marzo 2023. [En línea]. Available: <https://www.simplilearn.com/tutorials/docker-tutorial/what-is-dockerfile>. [Último acceso: 5 Julio 2023].

- [26] D. Pavlovic, «hp,» 18 Septiembre 2020. [En línea]. Available: <https://www.hp.com/us-en/shop/tech-takes/what-is-web-hosting>. [Último acceso: 12 Junio 2023].
- [27] M. Osman, «KINSTA,» 10 Mayo 2023. [En línea]. Available: <https://kinsta.com/es/blog/que-es-el-alojamiento-web/>. [Último acceso: 13 Junio 2023].
- [28] A. Schiaffarrino, «Infranetworking,» 15 Noviembre 2018. [En línea]. Available: https://blog.infranetworking.com/panel-control-web-hosting/#Administracion_y_Paquetes. [Último acceso: 10 Julio 2023].
- [29] I. Marca, «Server Mania,» Hosting Empowered, 10 Enero 2020. [En línea]. Available: <https://blog.servermania.com/select-web-hosting-control-panel>. [Último acceso: 12 Julio 2023].
- [30] «Reckcontent | Blog,» 5 Diciembre 2019. [En línea]. Available: <https://rockcontent.com/es/blog/cpanel-hosting/#:~:text=cPanel%20es%20una%20plataforma%20que,los%20niveles%20en%20un%20entorno..> [Último acceso: 12 Julio 2023].
- [31] «Cpanel,» Webpros, [En línea]. Available: <https://cpanel.net/products/cpanel-whm-features/>. [Último acceso: 16 Julio 2023].
- [32] «TecMint,» 20 Enero 2022. [En línea]. Available: <https://www.tecmint.com/install-webmin-in-linux/>. [Último acceso: 18 Julio 2023].
- [33] «Virtualmin,» [En línea]. Available: <https://www.virtualmin.com/>. [Último acceso: 19 Julio 2023].
- [34] «Webmin,» 1 Abril 2015. [En línea]. Available: https://doxfer.webmin.com/Webmin/Perl_Modules. [Último acceso: 26 Julio 2023].

- [35] C. G. Fernandez, «Observatorio Tecnológico,» 2006. [En línea]. Available: <http://recursostic.educacion.es/observatorio/web/ca/software/software-general/351-carlos-garcia-fernandez>. [Último acceso: 9 Agosto 2023].

7 ANEXOS

ANEXO I: Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 28 de agosto de 2023

De mi consideración:

Yo, FERNANDO VINICIO BECERRA CAMACHO, en calidad de Director del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE SERVICIOS DE RED CON HERRAMIENTAS DE DEVOPS elaborado por la estudiante ADRIANA SAMANTA CHALAVARRI PIARPUEZAN de la carrera en TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 12%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

https://epnecuador-my.sharepoint.com/:b:/g/personal/fernando_becerrac_epn_edu_ec/EfZ1RHJoBE9Mq51ME3L_dwQBhj_WPAp9zL9jZQ9yyqWvhQ?e=sjq4pe

Atentamente,

Fernando Vinicio Becerra Camacho

Docente

Escuela de Formación de Tecnólogos

ANEXO II: Código QR de la implementación y pruebas de funcionamiento



ANEXO III: Códigos Fuente

Primera instrucción.

```
FROM centos:7
```

Segunda instrucción.

```
RUN yum update -y && yum install -y \
```

Herramienta de descarga

```
wget \
```

Herramienta de descompresión de archivos

```
unzip \
```

Gestión de claves y certificados digitales

```
openssl \
```

Módulos Perl

Automatización de tareas

```
perl \
```

```
perl-IO-Tty \
```

Establece la comunicacion entre el usuario y webmin

```
perl-Net-SSLeay \
```

Bibliotecas de autenticación

```
perl-Authen-PAM \
```

Integridad y autenticidad de archivos y componentes

```
perl-Digest-MD5 \
```

```
perl-Digest-SHA \
```

interpretacion de datos

```
perl-Encode-Detect \
```

Construcción y depuración de paquetes con scripts Perl

perl-Data-Dumper

Descarga e instalación de Webmin

RUN wget http://www.webmin.com/download/rpm/webmin-current.rpm

RUN rpm -Uvh webmin-current.rpm

Tercera instrucción.

Especificación de puerto Webmin

EXPOSE 10000

Cuarta instrucción.

Inicio del servidor web Webmin y monitoreo del mismo

CMD /etc/webmin/start && tail -F /var/webmin/miniserv.log