Universidad Nacional Federico Villarreal

VRIN | VICERRECTORADO DE INVESTIGACIÓN

ESCUELA UNIVERSITARIA DE POSGRADO

DESPLIEGUE DE CONTENEDORES BASADO EN UNA NUEVA METODOLOGÍA PARA MEJORAR EL PROCESO DE PROVISIÓN DE SERVICIOS WEB EN LA UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO

Línea de investigación:

Sistemas de información optimización

Tesis para optar el Grado Académico de Doctor en Ingeniería de Sistemas

Autor:

Carrasco Poblete, Edwin

Asesor:

Soto Soto, Luis

(ORCID: 0000-0002-3799-645X)

Jurado:

Mujica Ruiz, Oscar Hugo

Marin Machuca, Olegario

Flores Masías, Edward José

Lima - Perú

2023

Reporte de Análisis de Similitud

Archivo:	1A_CARRASCO_POBLETE_EDWIN_DOCTORADO_2023.docx
Fecha del Análisis:	30/03/2023
Analizado por:	Astete Llerena, Johnny Tomas
Correo del analista:	jastete@unfv.edu.pe
Porcentaje:	1 %
Título:	DESPLIEGUE DE CONTENEDORES BASADO EN UNA NUEVA METODOLOGÍA PARA MEJORAR EL PROCESO DE PROVISIÓN DE SERVICIOS WEB EN LA UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
Enlace:	https://secure.urkund.com/old/view/155594627-195926-372377#q1bKLVayio7VUSrOTM/LTMtMTsxLTlWyMqgFAA==

UNIVERSIDAD NACIONAL FEDERICO VILLARREAL
OFICINA DE GRADOS Y GESTIÓN DEL EGRESADO
ESCUELA UNIVERSITARIA DE POSGRADO

DRA. MIRIAM LILIANA FLORES CORONADO
JEFA DE GRADOS Y GESTIÓN DEL EGRESADO

ESCUELA UNIVERSITARIA DE POSGRADO

DESPLIEGUE DE CONTENEDORES BASADO EN UNA NUEVA METODOLOGÍA PARA MEJORAR EL PROCESO DE PROVISIÓN DE SERVICIOS WEB EN LA UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO

Línea de Investigación:
Sistemas de información optimización

Tesis para optar el Grado Académico de
Doctor en Ingeniería de Sistemas

Autor
Carrasco Poblete, Edwin

Asesor
Soto Soto, Luis
ORCID: 0000-0002-3799-645X

Jurado
Mujica Ruiz Oscar Hugo
Marin Machuca Olegario
Flores Masías Edward José

Lima – Perú
2023

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS.

ÍNDICE DE TABLAS.

RESUMEN

Las tecnologías de información y comunicaciones tienen un rol crucial en el desarrollo de la sociedad moderna y para maximizar el beneficio de su uso, es imprescindible combinar tecnologías, técnicas y herramientas de manera eficiente para su explotación y para la puesta a disposición de la sociedad de plataformas que permitan desplegar servicios que coadyuven a ampliar y mejorar las opciones de acceso y disponibilidad tanto a servicios ya existentes como a servicios nuevos. El campo de las tecnologías de información y comunicaciones es dinámico, permanentemente surgen tecnologías innovadoras y muchas veces el tiempo de maduración de las mismas puede tomar años, postergando su adopción en entornos de producción. Parte de la madurez de una tecnología implica la creación de un ecosistema que la haga sostenible en el tiempo. Es el caso de la tecnología de contenedores: una tecnología innovadora eficiente para el despliegue de servicios de red que se presenta como una alternativa a la tecnología de virtualización utilizada hasta entonces. En el presente trabajo de tesis, se plantea una metodología para el uso de la tecnología de contenedores en el proceso de despliegue de servicios de red con el propósito de utilizar con mayor eficiencia los recursos de cómputo de las organizaciones. La metodología propuesta MFDC (Metodología Formal de Despliegue de Contenedores) integra las recomendaciones de seguridad para la gestión de contenedores propuestas por NIST. La metodología se ha validado con un caso de prueba que ha demostrado su viabilidad.

Palabras clave: contenedores, despliegue, Docker, metodología, máquinas virtuales.

ABSTRACT

Information and communication technologies play a crucial role in the development of modern society and to maximize the benefits of their use, it is essential to combine technologies, techniques and tools efficiently for their exploitation and for the provision of information to society. platforms that allow the deployment of services that help to expand and improve access and availability options for both existing services and new services. The field of information and communication technologies is dynamic, innovative technologies constantly emerge and many times their maturation time can take years, postponing their adoption in production environments. Part of the maturity of a technology implies the creation of an ecosystem that makes it sustainable over time. This is the case of container technology: an efficient innovative technology for the deployment of network services that is presented as an alternative to the virtualization technology used until then. In this thesis work, a methodology is proposed for the use of container technology in the process of deploying network services in order to use the computing resources of organizations more efficiently. The proposed MFDC methodology (Formal Container Deployment Methodology) integrates the security recommendations for container management proposed by NIST. The methodology has been validated with a test case that has demonstrated its feasibility.

Keywords: containers, deployment, Docker, methodology, virtual machines

I. INTRODUCCIÓN

1.1. Planteamiento del Problema

La tecnología de contenedores ha surgido de manera disruptiva y en un corto periodo de tiempo se ha posicionado como la plataforma preferida para el despliegue de servicios de red y aplicaciones en campos como la inteligencia artificial, computación de alto rendimiento, despliegue de aplicaciones basadas en micro servicios y otros.

Su adopción se debe a que permite utilizar de manera más eficiente la capacidad de procesamiento de los computadores y a su idoneidad para el despliegue rápido de servicios de la fase de desarrollo a la de producción.

Un contenedor se define como un entorno de ejecución que limita el ámbito de los recursos a los que un programa puede acceder (Nickoloff & Kuenzli, 2019). Este entorno de ejecución incluye la aplicación que se va a desplegar, las librerías requeridas por dicha aplicación, así como parámetro para su ejecución, empaquetadas de tal forma que pueden ejecutarse sin problemas de dependencia en cualquier plataforma que soporte esta tecnología.

Un administrador de red puede configurar un contenedor para un servicio de red y una vez que culmina con esa tarea, puede desplegar el mismo sin necesidad de hacer mayores configuraciones, lo que garantiza un despliegue rápido.

Los contenedores se almacenan en repositorios desde los cuales pueden instanciarse.

Los contenedores incluyen solamente los componentes estrictamente necesarios para la provisión de un servicio, lo que permite crear imágenes de contenedores muy compactas.

El tamaño reducido de un contenedor, permite la descarga rápida de la imagen desde el repositorio e igualmente el rápido despliegue en producción. De esta forma, en caso que se tuviese problemas con la operación de un contenedor, el servicio que este provee puede restablecerse rápidamente, mejorando la disponibilidad del servicio desplegado.

Históricamente, la provisión de servicios de red ha utilizado distintos enfoques (Jayaswal, 2006) y estrategias de optimización:

En un principio, debido a los costos de los servidores, se solía utilizar un solo servidor para proveer más de un servicio. Si bien, este enfoque permitía hacer un uso más eficiente de la infraestructura de servidores, la capacidad de gestión de conexiones podía convertirse en un problema con el paso del tiempo y el incremento de accesos a los servicios ofrecidos.

Otro problema relacionado con este enfoque es que el servidor podía convertirse en un cuello de botella y era, además, un único punto de falla.

Un enfoque posterior, posible debido al abaratamiento de los equipos servidores, consistió en utilizar un servidor por servicio. Con este enfoque, si fallaba un servidor, el resto de servicios podía ser aun accesible, con lo que mejoraba la disponibilidad de la red. Sin embargo, la eficiencia de uso de los servidores podía ser pobre. Además, el consumo de energía de los servidores y los centros de datos que los alojaban se volvieron un problema (Gough et al., 2015).

Para mitigar los problemas de uso eficiente de servidores y consumo alto de energía en un contexto de continuo descenso en los costos de hardware; se promovió el uso de tecnologías de virtualización. En este enfoque se virtualiza el hardware de los servidores y en cada uno de estos equipos virtualizados se despliega un servicio de red (siguiendo con el enfoque de un servidor – un servicio). Un servidor puede alojar múltiples máquinas virtuales y con la tecnología adecuada, puede desplegarse servicios redundantes, con alta disponibilidad y mejora en la eficiencia de consumo de recursos y energía.

Una limitación en la tecnología de virtualización descrita es que esta virtualiza solo el hardware de un computador sobre el que aun debía instalarse un sistema operativo y las herramientas y soporte necesario para desplegar el servicio de red ofrecido.

En términos de despliegue de las aplicaciones que proveen el servicio de red, no ocurre ningún cambio: las aplicaciones y servicios aún deben desplegarse en los servidores, los cuales luego pueden clonarse.

En el enfoque basado en la tecnología de contenedores, se busca mejorar la eficiencia de uso de los recursos de los servidores y hacer más flexible el proceso de despliegue de las aplicaciones.

Con la tecnología de contenedores, se prescinde de la necesidad de virtualizar el hardware de un servidor y, como se indicó previamente, se crea un entorno de ejecución independiente para una aplicación, el cual se encapsula como una entidad autocontenida que luego puede instanciarse a voluntad.

Esta entidad auto contenida puede asumirse como una máquina virtual ligera que aísla la aplicación que contiene de otras aplicaciones o contenedores que estuviesen ejecutándose en el mismo servidor.

Si algún problema afectase a un contenedor, los demás contenedores no se verán afectados por tal problema. Esta propiedad hace de los contenedores una tecnología fiable y los servicios ofertados sobre la misma: de alta disponibilidad.

Además de sus características de aislamiento y uso eficiente de recursos, los contenedores han ayudado a mejorar los procesos de desarrollo y despliegue de servicios de red por las características que se indican a continuación:

- Cuando se crea un contenedor, este puede consistir solo las aplicaciones y librerías suficientes para asegurar la correcta operación del servicio.
- Una vez creado un contenedor, este puede almacenarse como una imagen en un repositorio.
- Cuando se ejecuta un contenedor, este opera con todos los parámetros definidos en tiempo de creación.

- Se puede desplegar múltiples instancias de un contenedor y siempre operará de acuerdo a su configuración inicial
- Si un contenedor se modifica, estos cambios se almacenan como una nueva imagen.

Las capacidades descritas ayudan en los procesos de gestión de los servicios de red, al hacer más fácil la resolución de problemas.

1.2. Descripción del problema

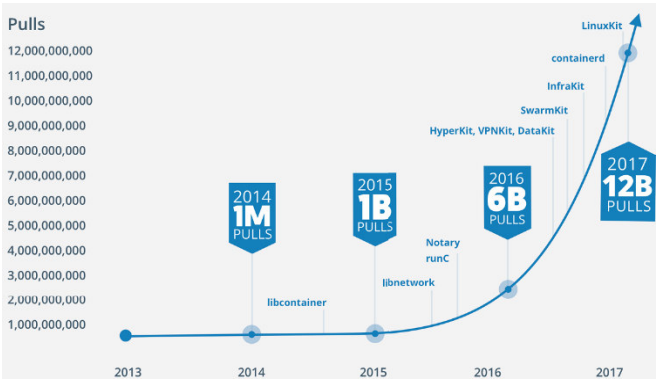
1.2.1. Ámbito Mundial

Oficialmente, la era de los contenedores comienza con el lanzamiento de Docker, por parte de la empresa Docker Inc. el año 2013. Desde entonces, ha sido adoptado a un ritmo acelerado, debido a las capacidades ya descritas.

En el estudio realizado por (Zhu & Bayley, 2018), se muestra que el número de descargas de imágenes de contenedores ha alcanzado la cifra de 12 000 000 000.

Figura 1

Número de descargas de imágenes de contenedores

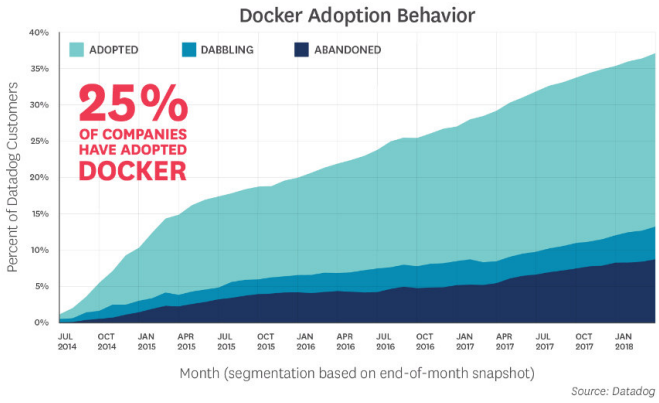


En el estudio (Khandhar & Shah, 2019), se reporta un comportamiento en la adopción de Docker del 25% y se afirma que después de 10 meses de la adopción de la tecnología, la

cantidad de contenedores desplegados se quintuplica. Estos datos se ilustran en la siguiente figura:

Figura 2

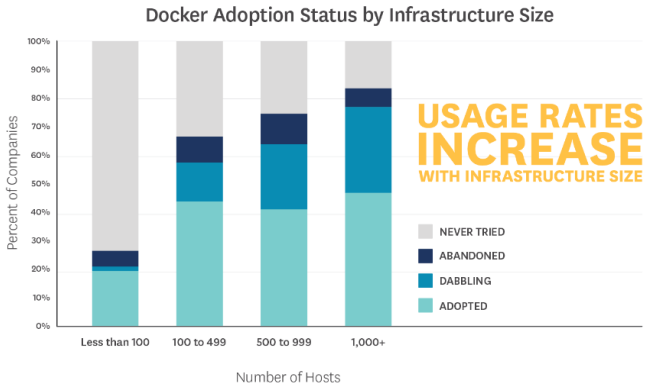
Patrón de adopción de Docker



En el reporte elaborado por (Datadog.com, 2018), se muestra el estado de adopción de Docker en función del tamaño de la infraestructura de las empresas consultadas (10 000). En este estudio se evidencia que el nivel de adopción de la tecnología de contenedores es mayor cuando la infraestructura es mayor (ver figura 3):

Figura 3

Estado de adopción de Docker por tamaño de infraestructura



1.2.2. Ámbito Nacional

En el ámbito nacional, el uso de contenedores no es aún extendido, aunque sí se viene utilizando como plataforma para el despliegue de aplicaciones web basadas en microservicios como, por ejemplo, en el trabajo de tesis (Serrano & Huallpamaita, 2018). En el ámbito comercial, la empresa Innovar SCR Ltda. desarrolla y provee servicio web utilizando la tecnología de contenedores Docker y software de orquestación Kubernets para el despliegue de las mismas en un ámbito global

1.2.3. Ámbito Local

La UNSAAC (Universidad Nacional de San Antonio Abad del Cusco) es una universidad pública, que fue fundada el primero de marzo de 1692 (UNSAAC, 2020). Ofrece 45 carreras profesionales de pre grado, organizadas en 10 facultades. Así mismo, cuenta con una escuela de postgrado en la que se ofrece formación especializada en maestrías y doctorados. Cuenta también con centros de prestación de servicios y centros de producción.

La población estudiantil a la que presta servicios la UNSAAC, de acuerdo al último boletín oficial (UNSAAC, 2017) asciende a 74 275 como se indica en la siguiente tabla:

Tabla 1

Población estudiantil UNSAAC

DEPENDENCIA	N° DE ESTUDIANTES
Pregrado	18 836
Centro de Idiomas	42 150
Instituto de Sistemas	1 513
CEPRU	11 776
TOTAL	74 275

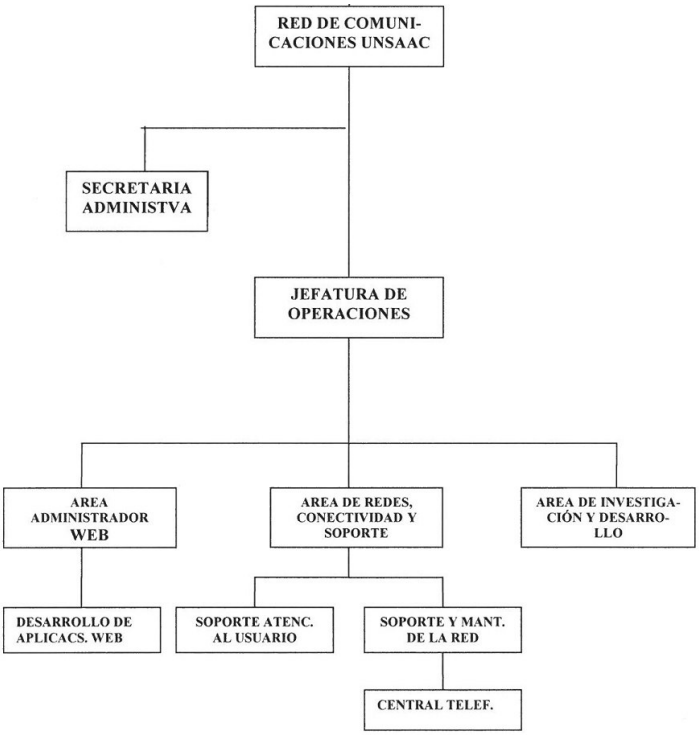
La información sobre los servicios que presta la UNSAAC, tales como las carreras profesionales ofertadas, el plan de estudios de cada una de ellas, los docentes de cada

departamento académico normas, el acervo bibliográfico, eventos académicos y muchos otros, se difunde a través del portal web de la institución: www.unsaac.edu.pe y es de importancia que los servicios de acceso a datos e información sean adecuados en términos de disponibilidad de los servicios, calidad de acceso y actualidad de la información.

La dependencia responsable de la gestión de servicios de red es la Red de Comunicaciones UNSAAC (RCU). La RCU depende administrativamente de la Dirección de Sistemas de Información y cuenta con personal organizado, según su organigrama, en la estructura descrita en la siguiente figura:

Figura 4

Organigrama de la RCU



Las funciones del personal de la RCU están establecidas en su manual de organización y funciones (MOF) (UNSAAC, 2009).

De acuerdo al MOF de la RCU, el responsable de la gestión de los servicios de red es el administrador web, siendo sus funciones específicas:

- Elaborar proyectos Web.
- Elaborar, actualizar y mantener la página Web principal de la UNSAAC y de las Dependencias Académicas y Administrativas de la Institución.
- Brindar información de eventos, conferencias y seminarios entre otros publicados en la página principal.
- Alcanzar a la Jefatura de Operaciones la información mensual de los avances y estado actual del Área.
- Verificar el adecuado funcionamiento de las aplicaciones Web montadas en el servidor Web.
- Facilitar la distribución de información, así como su fácil localización.
- Brindar sub-dominios para dependencias de la UNSAAC (Oficinas, carreras profesionales y centros de producción) y llevar el control.
- Administrar los servidores de Base de Datos y Web.
- Realizar Backups del servidor Web en forma continua.
- Realizar el mantenimiento a los pozos a tierra instalados por la RCU.
- Otras funciones afines que encomiende el Jefe de Operaciones.

A. Infraestructura y equipamiento de la RCU. Para proveer los servicios de red, la RCU cuenta con un cuarto de equipos ubicado en el sótano del edificio de la biblioteca central, closets de telecomunicaciones desplegados en los distintos pabellones del campus de Perayoc, red dorsal de fibra óptica y para la conexión de los usuarios finales: cableado horizontal UTP de categoría 6 y enlaces inalámbricos WiFi.

El cuarto de equipos de la RCU, entre otros, alberga:

- Switches core, desplegados para asegurar redundancia y conectados en una topología colapsada de 2 capas con los switches de acceso.
- Un servidor de telefonía IP.
- Servidores Blade para el alojamiento de las aplicaciones de red.

B. Servicios de red provistos por la RCU. Los servicios de red provistos por la RCU son:

- Servicio de nombres de dominio (unsaac.edu.pe y subdominios)
- Servicio web institucional y de dependencias administrativas y académicas

Los servicios indicados se ejecutan en máquinas virtuales desplegadas en servidores Blade de acuerdo al detalle descrito en la siguiente tabla:

Tabla 2

Máquinas virtuales desplegadas en la RCU

VM	OS	SERVICIO
VM_Acred_Cannon	Ubuntu Linux	Portal de proyecto de investigación
VM_Admision	Ubuntu Linux	Servicio Web para la Oficina de Admisión de la UNSAAC
VM_DNS	Debian GNU/Linux 5	Servicio de Nombres de dominio
VM_DSPACE	Ubuntu Linux	Servidor de la aplicación DSpace
VM_Moodle	Ubuntu Linux	Servidor para la plataforma Moodle
VM_RCU_Java	Ubuntu Linux	Servidor para aplicaciones Java
VM_Sbd	Ubuntu Linux	Servidor de bases de datos
VM_SWeb	Ubuntu Linux	Servidor Web institucional
VM_SWeb2	Ubuntu Linux	Servidor Web de pruebas
VM_Vrin	Ubuntu Linux	Servidor Web del VRIN

Los parámetros de operación de las máquinas virtuales se resumen en la siguiente tabla:

Tabla 3

Parámetros de máquinas virtuales desplegadas en la RCU

VM	CPUs	Memory	NICs	Disks	Video RAM KB	Provisioned MB
VM_Acred_Cannon	2	2,048	1	1	4,096	104,448
VM_Admision	2	8,192	1	1	4,096	22,528
VM_DNS	4	8,192	1	1	4,096	23,552
VM_DSPACE	2	4,096	1	1	4,096	34,817
VM_Moodle	2	4,096	1	1	4,096	55,296
VM_RCU_Java	2	4,096	1	1	4,096	14,336
VM_Sbd	2	4,096	1	1	4,096	24,576
VM_SWeb	4	4,096	1	1	4,096	34,817
VM_SWeb2	2	4,096	1	1	4,096	39,936
VM_Vrin	2	4,096	1	1	4,096	55,296

C. Proceso de provisión de servicios web. De acuerdo a información recopilada mediante encuesta al personal de la RCU, el proceso de provisión de servicios web inicia cuando el sitio web que se va a desplegar está concluido.

Las actividades de despliegue son:

1. Subir el servidor web y servidor de base de datos donde se encontrará alojado.
2. Crear el virtual host.
3. Crear subdominio para poner en marcha el sitio.
4. Si el sitio web debe almacenarse en un servidor independiente, se asigna una dirección IP antes de realizar los pasos previamente descritos.

Cabe resaltar que el proceso descrito no está formalmente establecido en documentación oficial, aunque se maneja documentación interna.

En cuando a la aplicación de normas para el proceso de provisión de servicios de red, se indica que no se utiliza ninguna.

Los aspectos de aseguramiento de la provisión de servicios de red comprenden:

- Gestión de puertos de red.
- Segmentación y filtrado de redes.

Sin embargo, estos procedimientos no están formalmente establecidos ni se sustentan en normas.

Para la evaluación del proceso de provisión de servicios web utilizaremos dos indicados para medir el nivel de servicio alcanzado con la aplicación de la tecnología de contenedores y dos indicadores asociados al consumo de recursos utilizados por los servicios web:

1. Tiempo de respuesta: Mide el tiempo desde que arriba una petición HTTP al servidor hasta que el servidor devuelve una respuesta HTTP (Subraya, 2006)
2. Rendimiento: Mide el número de peticiones que puede atender el servidor web por unidad de tiempo (Subraya, 2006).
3. Uso de CPU: Mide el tiempo de CPU utilizado por una tarea en relación a la suma de dicho tiempo más el tiempo de CPU utilizado para otra tarea distinta (Subraya, 2006)
4. Uso de memoria: Mide la cantidad de memoria utilizada por una tarea en relación a la cantidad total de memoria disponible en el sistema (Subraya, 2006)

1.3. Formulación del problema.

PROBLEMA GENERAL

¿En qué medida mejorará el proceso de provisión de servicios de red en la UNSAAC al desplegar la tecnología de contenedores utilizando la metodología propuesta Metodología Formal de Despliegue de Contenedores (MFDC)?

PROBLEMAS ESPECÍFICOS

1. ¿En qué medida disminuirá el tiempo de respuesta del servicio web al utilizar tecnología de contenedores?
2. ¿En qué medida incrementará el rendimiento del servicio web al utilizar tecnología de contenedores?
3. ¿En qué medida disminuirá el uso del CPU del servicio web al utilizar tecnología de contenedores?
4. ¿En qué medida disminuirá el uso de memoria del servicio web al utilizar tecnología de contenedores?

1.4. Antecedentes.

1.4.1. Internacionales

1. "*Docker Container Service for Microservice Applications*" (Balakumar & Kavitha, 2018) conducen un estudio sobre el uso de contenedores Docker para aplicaciones basadas en microservicios. Esta experiencia da sustento al uso de contenedores para la provisión de servicios web, con instalaciones que solo contengan el propio servidor web y el contenido ofrecido.
2. "*Comparison of Windows and Linux as Docker Hosts*" (Pfaller & Hartley, 2018). Evalúa las plataformas Microsoft Windows y Linux para el despliegue de la tecnología de contenedores Docker. Se utilizará para sustentar el nivel de madurez logrado por esta tecnología.
3. "*Performance Evaluation of Microservices Architectures using Containers*". (Amaral et al., 2015). Evalúa el uso de contenedores en el despliegue de microservicios proponiendo dos modelos: maestro – esclavo y contenedores anidados. Se utilizará para evaluar las opciones de despliegue de contenedores

1.4.2. Nacionales

1. “Arquitectura para el desarrollo e implementación de servicios web” (Huanca, 2017). Evalúa el despliegue de servidores web utilizando contenedores Docker, concluyendo: … “

   - El desarrollo de aplicaciones web utilizando el modelo Arquitectónico para el desarrollo de Aplicaciones Web propuesto, mediante el estilo de Microservicios, resulta beneficioso ya que permite tener entregas continuas del producto, y que además proporciona reproductibilidad, flexibilidad y reusabilidad del módulo por otros componentes del sistema, tanto internos como externos; permitiendo la integración con metodologías ágiles de desarrollo de software.
   - La utilización de la arquitectura propuesta para la Implementación y despliegue de aplicaciones web, mediante la tecnología de contenedores Docker, reduce ampliamente los tiempos empleados de la forma tradicional, generando también escalabilidad tanto en centro de datos privados y computación en la nube, notándose la eficiencia del despliegue.
   - El desarrollo y despliegue de los Servicios Web RESTful en las aplicaciones desarrolladas en el caso de estudio presentaron reducción de tiempo en preparar ambientes de desarrollo y puestas de producción, ya que las tecnologías utilizadas son las más adecuadas para el despliegue en centros de datos privados y servicios en Computación en la Nube, y servicios prestados como EC2 de Amazon Web Service, lo cual reduce la brecha en el uso do de estas nuevas tecnologías de información.”

Si bien el trabajo se orienta al uso de contenedores en el ámbito de las aplicaciones web; este no incluye las actividades orientadas a la provisión de servicios en una plataforma propia como es el caso de la RCU

1.5. Justificación de la investigación.

El presente trabajo se justifica desde las siguientes perspectivas:

Aporte teórico

El presente trabajo de investigación propondrá una metodología inédita para el despliegue de contenedores. La metodología propuesta especificará las tareas, procesos y documentación formal requerida para las actividades de gestión y despliegue de contenedores

Aporte práctico

La metodología propuesta será puesta en práctica en la RCU – UNSAAC, para validar la misma y para resolver problemas de la realidad de la institución.

Aporte tecnológico

El presente trabajo aplicará la tecnología de contenedores para el despliegue de servicios de red en la UNSAAC. Su aplicación servirá de referencia para proyectos futuros en organizaciones que provean servicios de red similares.

1.6. Limitaciones de la investigación.

- Cambios constantes en el campo del desarrollo y aplicaciones de la tecnología de contenedores que puede afectar la estabilidad del trabajo de investigación.
- Bibliografía sobre diseño de metodologías escasa.

1.7. Objetivos.

OBJETIVO GENERAL

Desplegar contenedores basado en la metodología MFDC para mejorar el proceso de Provisión de Servicios de Red en la Universidad Nacional de San Antonio Abad del Cusco

OBJETIVOS ESPECÍFICOS

1. Reducir el tiempo de respuesta del servicio web al utilizar la tecnología de contenedores.
2. Incrementar el rendimiento del servicio web al utilizar la tecnología de contenedores.
3. Reducir el uso de CPU del servicio web al utilizar la tecnología de contenedores.
4. Reducir el uso de memoria del servicio web al utilizar la tecnología de contenedores.

1.8. Hipótesis.

HIPÓTESIS GENERAL

El despliegue de contenedores basado en la metodología MFDC mejora el proceso de provisión de servicios de red en la UNSAAC

HIPÓTESIS ESPECÍFICAS

H1: El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC reduce el tiempo de respuesta del servicio web.

H2: El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC incrementa el rendimiento del servicio web.

H3: El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC reduce el uso de CPU del servicio web.

H4: El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC reduce el uso de memoria del servicio web.

II. MARCO TEÓRICO

2.1. Contenedores

Un contenedor es una agrupación de aplicaciones y sus dependencias, encapsuladas como una unidad de software estándar, que puede ejecutarse en un espacio o contexto de ejecución auto descrito, portable (Open Container Initiative, 2021) y aislado de otras aplicaciones que pudieran estar ejecutándose en el mismo computador.

El aislamiento de los contenedores se funda en las tecnologías de espacios de nombres (Namespaces) y grupos de control (control groups o cgroups).

Las aplicaciones, sus dependencias e información de configuración que definen un contenedor, pueden guardarse como una imagen persistente. Propiamente, la instancia de tal imagen es la que toma el nombre de contenedor.

Puesto que la imagen contiene todos los recursos que las aplicaciones componentes del contenedor requieren; la instanciación de una imagen de contenedor, podrá ejecutarse sin problemas en cualquier computador que soporte esta tecnología.

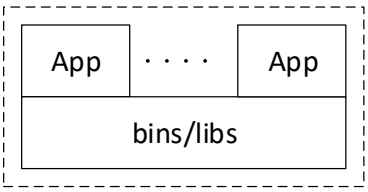
Esta última capacidad, permite que los contenedores sean portables: puede crearse una imagen persistente de una aplicación y sus dependencias y luego desplegarse, sin necesidad de realizar ajustes, en cualquier computador.

Más aun, múltiples instancias de una misma imagen pueden ejecutarse en un computador, sin interferir entre sí, puesto que cada una de las instancias se ejecutará en su propio entorno de ejecución, asignado y gestionado por el sistema operativo subyacente.

A. Arquitectura de contenedores. Lógicamente, un contenedor puede representarse como una agrupación de una o más aplicaciones y sus dependencias, encapsuladas como una entidad de ejecución atómica. De manera simplificada, este concepto se ilustra en la siguiente figura:

Figura 5

Representación conceptual de un contenedor

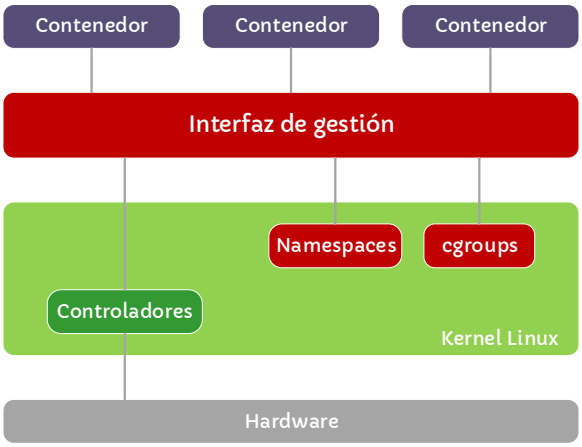


La atomicidad de un contenedor asegura que, si el contenedor se crea en un sistema, este pueda ser transportado a otro sistema sin perder funcionalidad. Esta característica posibilita utilizar la tecnología de contenedores para el despliegue de aplicaciones sin necesidad de procesos de instalación y eventuales problemas de compatibilidad de versiones de librerías de las plataformas sobre las que se ejecuta el contenedor.

La relación de un contenedor con otros componentes del sistema operativo Linux, se explica con el diagrama lógico adaptado de (Red Hat, 2021)

Figura 6

Relación de contenedores y sistema operativo Linux



La tecnología de Namespace, permite que los procesos puedan tener vistas diferentes del sistema subyacente (Ivanov, 2017). Si cada contenedor se ejecuta en su propio Namespace,

entonces este existirá aislado de otros procesos del sistema. Aun si comparten los mismos recursos hardware.

Los Namespaces vigentes son:

- Mount: provee una visión independiente del sistema de archivos a un proceso
- UTS – Unix Timesharing: aislamiento para el nombre de host y el nombre de dominio del host
- IPC – Interprocess Communication: aislamiento IPC y sincronización mediante semáforos mutex y otros.
- PID – Proces ID: aislamiento de identificadores de procesos dentro de un espacio de nombres.
- User: aislamiento de nombres de usuario y grupos dentro de un espacio de nombres
- Network: aislamiento de recursos de red dentro de un espacio de nombres. Cada contenedor puede tener su propia pila de comunicaciones de red, con dirección IP propia y diferente a la del host.

Con cgroups, se puede asignar recursos como tiempo de CPU, memoria y otros, a grupos de tareas definidos por el usuario, en cantidades también controlables por el usuario.

Algunos cgroups son:

- cpu: asigna tiempo de CPU a las tareas
- cpuset: asigna nucleos de CPU a las tareas
- memory: asigna memoria para las tareas
- blkio: limita acceso de E/S a los dispositivos
- devices: permite/deniega acceso a los dispositivos
- etc.

B. Principios de contenedores estándar. De acuerdo al documento "Open Container Initiative Runtime Specification", emitido por Open Container Initiative (OCI), un contenedor estándar, es una unidad estándar de despliegue de software que se fundamenta en 05 principios (Open Container Initiative, 2021):

1. Operaciones estándar: los contenedores estándar definen un conjunto de operaciones estándar.
2. Independencia del contenido: todas las operaciones sobre los contenedores tienen el mismo efecto, independientemente del contenido d ellos contenedores.
3. Independencia de la infraestructura: los contenedores estándar pueden ser ejecutados en cualquier plataforma soportada por OCI.
4. Diseñado para la automatización: los contenedores estándar se diseñan para la automatización, por cuanto ofrecen las mismas operaciones estándar, independientemente del contenido y la infraestructura.
5. Despliegue de grado industrial: tomando ventaja de las propiedades antes mencionadas, la tecnología de contenedores hace más fácil canalizar y automatizar las líneas de despliegue de software.

C. Ventajas de los contenedores. Las ventajas del uso de contenedores se pueden resumir en (Gutterman et al., 2019):

- Bajo consumo de hardware: un contenedor consume menos recursos de hardware en comparación a otras alternativas de similar propósito, como son las máquinas virtuales.
- Aislamiento de entorno: un contenedor se ejecuta en un entorno cerrado e incluye las librerías mínimas que requiere para su ejecución. Las actualizaciones que se apliquen a un contenedor, no afectan a otros contenedores.

- Despliegue rápido: puesto que, un contenedor es una unidad de despliegue autocontenida, su despliegue es rápido y no requiere de interrumpir ningún servicio del sistema operativo.
- Despliegue en entornos múltiples: por cuanto un contenedor incorpora todos los recursos que necesita para ejecutarse, este puede desplegarse en cualquier entorno que soporte la tecnología de contenedores. En la actualidad, sistemas operativos como Windows 10 y MacOS soportan esta tecnología.
- Reusabilidad: un contenedor puede reutilizarse sin necesidad de configurar un sistema operativo completo. Más aun, una imagen de contenedor puede servir de base para la creación de nuevas imágenes.

D. Estado de un contenedor. El estado de un contenedor se define por las siguientes propiedades (Open Container Initiative, 2021):

1. ociVersion: versión de la especificación de ejecución (Runtime) OCI.
2. id: Identificador del contenedor.
3. status: estado de ejecución del contenedor. Puede ser:
    a. creating
    b. created
    c. running
    d. stopped
4. pid: id del proceso contenedor
5. bundle: la ruta absoluta al directorio del contenedor
6. annotations: lista de anotaciones asociadas al contenedor

E. Ciclo de vida de un contenedor. El ciclo de vida de un contenedor OCI está determinado por la secuencia de eventos que se suceden, desde la creación de un proceso, hasta cuando este deja de existir. De acuerdo a (Open Container Initiative, 2021), el ciclo de vida comprende doce fases, las cuales se indican a continuación:

1. Se invoca el comando create, con una referencia a la ubicación del paquete y un identificador único.
2. El entorno de ejecución del contenedor se crea de acuerdo a la configuración descrita en el archivo config.json.
3. Se invocan los prestart hooks (mecanismos de gestión de eventos).
4. Se invocan los createRuntime hooks.
5. Se invocan los createContainer hooks.
6. Se invoca el comando start con el identificador único del contenedor.
7. Se invocan los startContainer hooks.
8. Se ejecuta un programa de usuario, de acuerdo a las especificaciones de process.
9. Se invocan los poststart hooks, si ocurren errores, el Runtime emite una advertencia, pero el ciclo de vida continúa como si el hook hubiera tenido éxito.
10. El proceso contenedor termina.
11. Se invoca el comando delete con el identificador único del contenedor.
12. El contenedor debe destruirse deshaciendo los pasos ejecutados en la fase de creación.
13. Se invocan los poststop hooks, si ocurren errores, el Runtime emite una advertencia, pero el ciclo de vida continúa como si el hook hubiera tenido éxito.

F. Formato de imagen de un contenedor OCI. El formato de un contenedor que cumple con las especificaciones OCI comprende (Open Container Initiative, 2021):

1. *Image Manifest*: documento de descripción de los componentes del contenedor.

2. *Image Index*: índice de manifiestos de imagen.
3. *Image Layout*: composición de sistema de archivos que representa los contenidos de una imagen.
4. *Filesystem Layer*: descripción de del sistema de archivos del contenedor.
5. *Image Configuration*: documento que determina el ordenamiento de niveles y configuración de la imagen para su traducción a un paquete de ejecución.
6. *Conversion*: descripción del proceso de traducción.
7. *Descriptor*: una referencia que describe el tipo, metadatos y dirección del contenido del contenido referenciado [sic].

2.2. Docker

Docker es una tecnología que comprende un conjunto de aplicaciones orientadas a facilitar la gestión de contenedores Linux para lo cual, Docker ofrece al usuario una capa de abstracción que lo libera de manejar los detalles técnicos de la gestión del ciclo de vida de los contenedores.

Con Docker, la facilidad con la que se puede construir, modificar, iniciar, ejecutar, detener y, en forma general, manipular contenedores es impresionante (Negus, 2016).

Las tecnologías en las que se basan los contenedores han existido y se han utilizado desde antes (El Amri, 2017): Chroot Jail, FreeBSD Jails, Linux-Vserver, Contenedores de Solaris, OpenVZ, Contenedores de procesos, LXC (Linux Containers). La característica de estas tecnologías es que aíslan programas en tiempo de ejecución, limitando su alcance y acceso al sistema de archivos y recursos estrictamente necesarios para su operación (Nickoloff & Kuenzli, 2019). Sin embargo, estas eran soluciones bastante complejas y propensas a errores de configuración.

Docker agrega a la tecnología de aislamiento de aplicaciones, herramientas de alto nivel que hacen más fácil su uso, aplicación y gestión (El Amri, 2017):

- Formato estandarizado para empaquetar aplicaciones y sus dependencias en un único objeto que puede ser transferido a cualquier máquina que soporte Docker y ejecutarse con la garantía que el entorno de ejecución será el mismo.
- Docker se optimiza para el despliegue de aplicaciones antes que de máquinas virtuales.
- Herramientas para automatizar el ensamblaje de un contenedor a partir de su código fuente.
- Gestión de versiones de contenedores.
- Reutilización de contenedores, tanto para despliegue como para crear nuevos contenedores en base a ya existentes.
- Repositorio de contenedores (https://registry.hub.docker.com).
- API para la automatización y personalización de la creación y despliegue de contenedores.

2.2.1. Arquitectura de Docker.

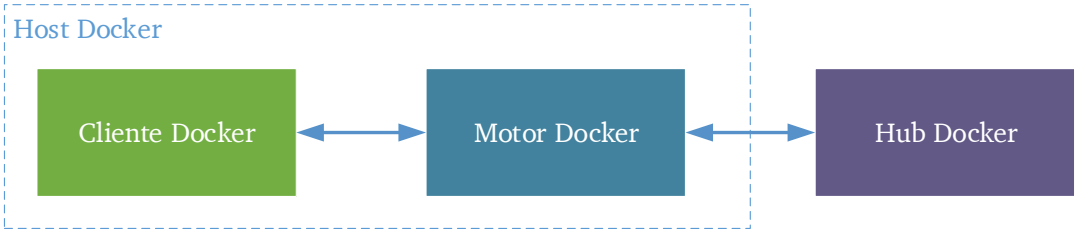
Docker está organizado alrededor de tres componentes fundamentales (Sabharwal & W, 2015):

1. El motor Docker
2. El hub Docker
3. El cliente Docker

Los cuales interactúan entre sí de la forma expuesta en el siguiente gráfico:

Figura 7

Arquitectura de Docker



Estos componentes cumplen las siguientes funciones:

A. Motor Docker (Docker Engine). Se compone a su vez de (Docker Inc., 2021):

- Un servidor denominado Docker daemon, cuya función es atender las peticiones del cliente Docker y gestionar los objetos Docker como imágenes, contenedores, redes y volúmenes.
- Una API (Application Programming Interface) REST (REpresentational State Transfer), que sirve de interfaz para la interacción de programas con Docker.
- Un cliente de línea de comandos (CLI), que canaliza las órdenes y respuestas entre el usuario y el demonio Docker.

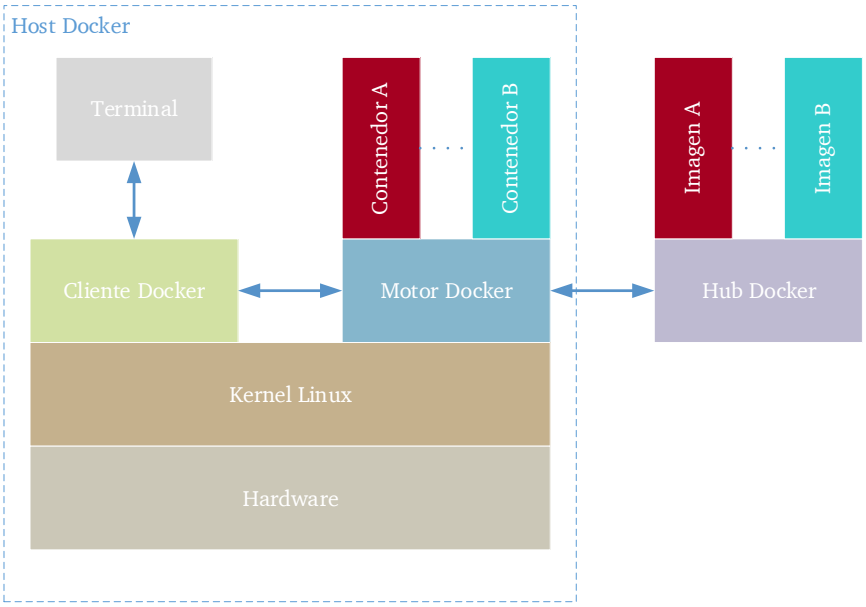
B. Hub Docker (Docker registry). Tiene como función almacenar imágenes de contenedores. Un hub Docker puede ser local, que es caso en el que las imágenes de contenedores se almacenan en el propio host Docker, o remoto, que es caso en el que el repositorio de imágenes se almacena en un servidor externo (v.g. https://hub.docker.com/).

C. Cliente Docker. Es el componente, que permite la interacción del usuario con el sistema de gestión de contenedores Docker. El cliente Docker recibe las peticiones del usuario a través de un terminal y las envía, utilizando las API REST, al Docker daemon.

Para mayor claridad, en el siguiente gráfico adaptado de (Docker Inc., 2021) y (Nickoloff & Kuenzli, 2019) se muestra los componentes de Docker y su lugar respecto de los demás componentes de un computador:

Figura 8

Docker y Linux



2.2.2. Imagen de contenedor Docker.

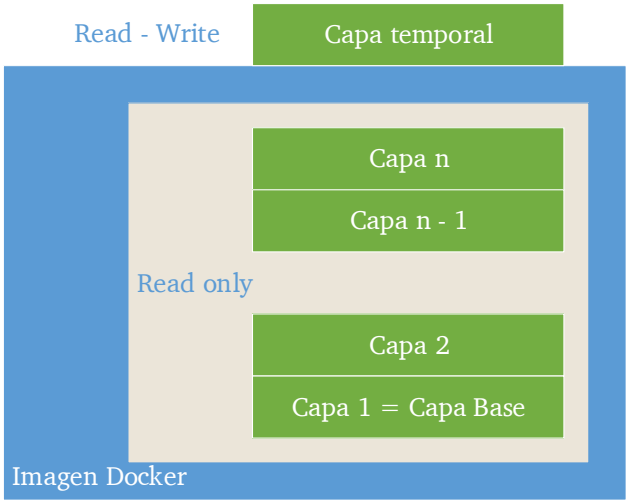
Una imagen de contenedor Docker (imagen Docker), es una plantilla de solo lectura que contiene instrucciones para la creación de contenedores.

Una imagen Docker es una asociación de varias capas, de solo lectura, combinadas como una entidad única (Poulton, 2020).

La primera capa se conoce como la capa base (Schenker, 2020). Las capas de una imagen son inmutables y cada capa adicional contiene solo los cambios respecto de la capa anterior. Todas las capas se combinan utilizando un sistema de archivos UFS (Union FileSystem)

Figura 9

Capas de una imagen Docker



Al instanciarse un contenedor, se agrega una capa temporal de lectura/escritura para las operaciones en tiempo de ejecución.

Si el contenedor se detiene, al instanciarlo nuevamente, el contenedor iniciará con las capas inmutables y se creará una nueva capa temporal. Es decir, un contendor siempre inicia con la imagen Docker en el mismo estado inicial (Schenker, 2020).

2.2.3. Contenedor Docker.

Un contenedor Docker es la instancia de una imagen de un contenedor. La relación entre una imagen y un contenedor es muy parecida a la relación entre una clase y un objeto en el paradigma orientado a objetos.

Así como un objeto es la instanciación de una clase; un contenedor es la instanciación de una imagen Docker.

El motor Docker es el responsable de la instanciación de un contenedor a partir de una solicitud del cliente Docker.

Al recibir la petición del cliente Docker, el motor Docker extrae una imagen Docker desde un hub Docker y lo instancia en el host Docker.

2.2.4. Dockerfile.

Dockerfile es un archivo que contiene instrucciones para la construcción de una imagen Docker.

Algunas instrucciones que se pueden incluir en el archivo Dockerfile son (Stoneman, 2020):

FROM – permite especificar una imagen base para la imagen Docker

RUN – permite ejecutar comandos Linux en el contenedor

EXPOSE – permite exponer un puerto en el contenedor

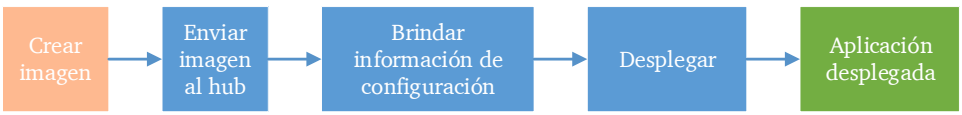
2.2.5. Flujo de despliegue Docker.

Con Docker, el despliegue de contenedores consiste de los siguientes pasos (Kane & Matthias, 2018):

1. Crear la imagen Docker
2. Enviar, opcionalmente, la imagen Docker a un hub Docker
3. Establecer detalles de configuración y proveer recursos
4. Desplegar contenedores basados en la imagen Docker

Figura 10

Flujo de despliegue Docker



2.2.6. Ventajas de Docker.

Las ventajas de utilizar Docker en el despliegue de aplicaciones han obligado a su adopción en plataformas como Microsoft Azure, AWS de Amazon, Alibaba Cloud, etc. por cuanto ofrece los siguientes beneficios (El Amri, 2017) (Miell & Sayers, 2016) (Turnbull, 2016):

- Despliegue rápido de aplicaciones.
- Distribución y colaboración de imágenes de contenedores.
- Multi tenencia (Multi tenancy), término que describe la capacidad de instanciar múltiples contenedores a partir de una imagen Docker (Krebs et al., 2012).
- Alta disponibilidad.
- Aislamiento.
- Arquitecturas de microservicios.
- Integración continua.
- Entrega continua.
- Escalabilidad.

2.3. Servicio Web

La World Wide Web (WWW), o simplemente la web, es un servicio de red creado por Tim Berners-Lee en 1994, que cambió la forma en que hasta ese entonces se utilizaba Internet (Kurose & Ross, 2013).

La WWW, comprende una infraestructura de información distribuida a nivel global, organizada y almacenada en servidores web e integrada mediante enlaces entre sí, accesibles a través de navegadores web.

En el núcleo de la WWW se encuentran dos elementos fundamentales: el servidor web y el cliente web, apoyados en tres estándares fundamentales:

1. HyperText Markup Language (HTML) – lenguaje que permite especificar el contenido y la organización de una página web.
2. Uniform Resource Locator (URL) – especifica el formato y semántica de los identificadores de una página web.
3. HyperText Transfer Protocol (HTTP) – especifica la interacción entre un cliente y un servidor web.

El servidor web almacena contenido multimedia accesible a través de una página web, definida utilizando el lenguaje de etiquetas HTML (Hyper Text Markup Language).

Una página HTML contiene referencias a objetos, los cuales pueden ser archivos de texto, imágenes, audio o video. Los objetos pueden ser referenciados mediante un URI (Uniform Resource Identifier), que en el caso de objetos web se conoce como URL (Uniform Resource Locator). Estos objetos pueden residir en el mismo servidor web o en servidores web remotos y distintos del servidor que aloja la página web (Shklar, 2009).

Un cliente web o navegador web, permite recuperar las páginas web, interpretar el archivo HTML y luego de interpretar las especificaciones contenidas en dicha página, recuperar los objetos web referenciados y representar localmente la página web y los archivos referenciados.

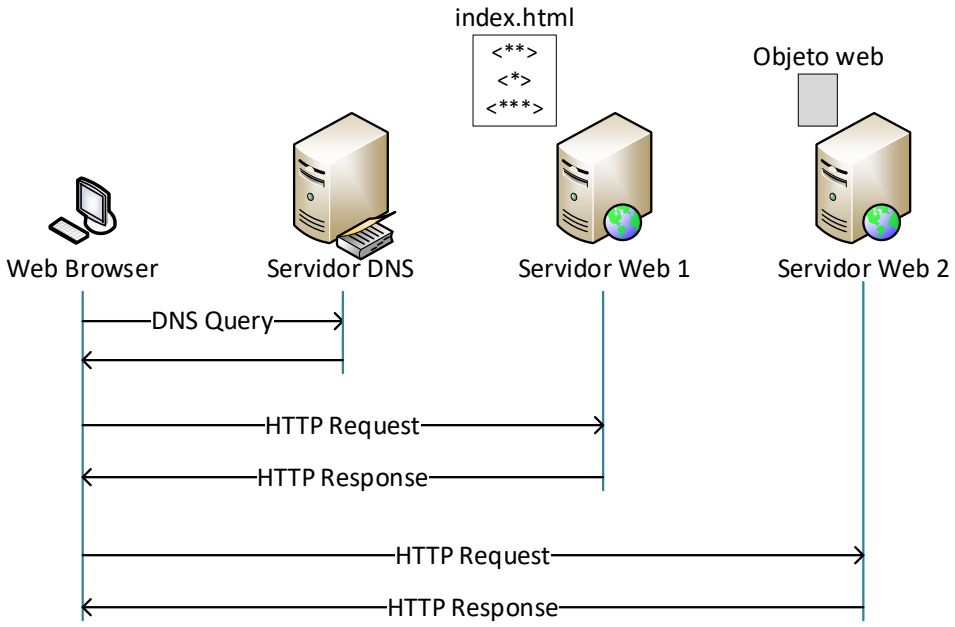
Para representar, localmente una página web, el cliente web interpreta las etiquetas de archivo HTML y las especificaciones de estilo definidas en un archivo CSS (Cascading Style Sheet).

Para la interacción entre el cliente y el servidor web se utiliza el protocolo HTTP (HyperText Transfer Protocol).

En forma simplificada, la arquitectura web y una posible sesión HTTP se muestra en el gráfico adjunto adaptado de (Kurose & Ross, 2013):

Figura 11

Arquitectura web y sesión HTTP



Un servidor web opera en el denominado sitio web; concepto que implica, además del servidor web, un nombre de dominio, por ejemplo: www.unsaac.edu.pe.

La tecnología web ha evolucionado desde sus inicios, en los que los contenidos proveídos eran de carácter estático, basados en textos e imágenes almacenadas, a aplicaciones web dinámicas con información generada ad – hoc mediante CGI (Common Gateway Interface) y, más recientemente, con contenido activo que permite ejecutar programas, en lenguajes de programación script que se descargan como parte de las páginas web, en el lado del cliente (Forouzan, 2012). Esta última capacidad permite la creación de aplicaciones web capaces de reemplazar las aplicaciones de escritorio tradicionales.

En entornos de producción, un servidor web debe estar diseñado para atender cientos o miles de conexiones por dos razones:

1. El sitio web puede ser referenciado directamente.

2. El sitio web puede ser referenciado por servidores de otros sitios web.

Las referencias entre servidores de sitios web crean una red de referencias, que en el argot tecnológico ha derivado en llamar a esta red la “World Wide Web” o telaraña mundial.

2.4. Ingeniería de métodos

La ingeniería de métodos, es, de acuerdo a (Rolland, 2007), la disciplina que estudia las técnicas de ingeniería para la construcción, aplicación, evaluación y gestión de métodos para el desarrollo de Métodos de Desarrollo de Sistemas de Información (ISDM – Information Systems Development Methods).

Dentro del ámbito de la ingeniería de métodos, un método situacional, es el resultado de la selección e integración de partes de ISDM en un nuevo método adaptado a una situación específica.

El enfoque de la ingeniería de métodos se funda en la asunción que ninguna metodología especifica puede resolver suficientes problemas y que, por lo tanto, las metodologías deben crearse específicamente para un conjunto de requerimientos en particular. A fin hacer esto de forma viable y económicamente efectiva, se ensamblan metodologías a partir de componentes de métodos existentes, aplicando los principios de modularidad y reusabilidad (Gonzales-Perez, 2007).

Los componentes de métodos existentes se denominan fragmentos de métodos, mientras que el proceso de combinar estos fragmentos en un método nuevo se denomina ensamblado de métodos (Brinkkemper et al., 1998).

A. Reglas del ensamblado de métodos. En (Brinkkemper et al., 1998), se plantean 12 reglas que restringen el ensamblado de métodos:

1. Por lo menos un concepto, asociación o propiedad debe introducirse a cada fragmento de método que se ensamblará, i.e. un fragmento de método que se ensamblará no debe ser el subconjunto de otro.
2. Debemos tener al menos un concepto o asociación que conecta dos fragmentos de métodos que se ensamblarán.
3. Si agregamos nuevos conceptos, debe haber conectores entre ambos fragmentos de métodos ensamblados.
4. Si agregamos nuevas asociaciones, los dos fragmentos de métodos ensamblados deben participar en él.
5. No existen partes aisladas en los fragmentos de métodos resultantes.
6. No hay conceptos que tienen el mismo nombre y que tienen diferentes ocurrencias en la descripción de un método.
7. La actividad de identificar los conceptos y relaciones agregados deben realizarse después que se ha identificado sus conceptos asociados.
8. Sean A y B dos fragmentos de métodos por ensamblar, y C el nuevo fragmento de método. En C, debemos tener al menos un producto que es la salida de A y que es la entrada de B o viceversa.
9. Cada fragmento de producto debe producirse por un fragmento de proceso correspondiente.
10. Suponga que un fragmento de producto se ensambló. El fragmento de proceso que produce este producto consiste de los fragmentos de proceso que producen los componentes del fragmento de producto.

11. Un fragmento de método técnico debe dar soporte a un fragmento de método conceptual.
12. Si existe una asociación entre dos fragmentos de productos, debe existir por lo menos una asociación entre sus respectivos componentes.

B. Proceso para la ingeniería de métodos contextuales y específicos de proyectos. El proceso para la creación de métodos contextuales comprende 4 pasos (Bucher et al., 2007):

1. Planificar o evaluar el método, considerando:
    a) No existe aún un método para una situación dada.
    b) Existe un método para una situación determinada.
2. Identificar factores de contexto y factores del tipo de proyecto, considerando el cuerpo de conocimientos actual correspondiente:
    a) Modelos, métodos, modelos de procedimientos y teorías sobre el artefacto del método existente.
    b) Conocimiento genérico sobre los modelos de procedimiento existentes y
    c) Experiencia de proyectos prácticos o ganada por observación.
3. Analizar contextos y tipos de proyectos para descubrir los métodos aplicables a un tipo de proyecto, de acuerdo a su contexto. Se sugiere utilizar una matriz Contexto vs. Tipo de Proyecto.
4. Construir y validar el método situacional. Los métodos resultantes pueden ser de los tipos:
    a) Método configurable para una situación específica.
    b) Método específico para la situación.
    c) Método valido para toda situación.

En cualquier caso, el método debe validarse mediante su aplicación en una situación del mundo real, para la cual se proclama válido.

2.5. Metodologías

El Diccionario de la Real Academia Española define una metodología como "ciencia del método" y "el conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal".

Por su parte el diccionario Merriam-Webster, dice que la metodología es "el cuerpo de métodos, reglas y postulados empleados en una disciplina: un procedimiento particular o un conjunto de procedimientos".

Una definición más amplia y pertinente, desde la perspectiva del autor del presente trabajo, es postulada en (Maguire, 2019), que plantea que "Una metodología no es un método o un grupo de métodos. Metodología es aquello que abarca la lógica de la elección de los métodos, que incluye qué paradigma, a qué base conceptual o teórica se recurre y cómo se relacionan los métodos con la elección conceptual, teórica y paradigmática y cómo los métodos se relacionan entre sí para lograr la evidencia y la fiabilidad requerida para hacer una contribución al conocimiento existente".

A. Características de las metodologías. Desde la perspectiva de (Maguire, 2019), una metodología se caracteriza por:

- Métodos, descritos como las herramientas de recolección de datos y el tipo de análisis empleado. Los métodos pueden ser cuantitativos, cuando los datos pueden ser medidos estadísticamente, o cualitativos, cuando los datos son de experiencias y perspectivas.
- Paradigma, que es determinado por cómo vemos la realidad.

- Fundamentos teóricos y conceptuales en los que se fundamenta la investigación.
- Fiabilidad, que caracteriza la confiabilidad de todos los aspectos de la investigación.

2.6. DevOps

Según (Reed, 2020); DevOps, término derivado de "development and operation", hace referencia a "... un conjunto de prácticas configuradas para combinar operaciones de TI con desarrollo de software para reducir el ciclo de vida de desarrollo de sistemas necesario para lograr software de alta calidad".

DevOps surge como una respuesta a la creciente complejidad del despliegue de aplicaciones que, no solo comprenden el software propio, sino, además, servicios en la nube, recursos de red, DNS, firewalls, etc.

Antes del surgimiento de DevOps, el software se producía en el departamento de desarrollo de software, siguiendo algún método de desarrollo de software. Por ejemplo, el modelo de cascada o más recientemente, los métodos de desarrollo ágiles.

Una vez completado el producto de software, este se entregaba al departamento de operaciones, el cual tenía la responsabilidad de desplegar el producto software en la plataforma de producción, operar el sistema en producción y monitorear su operación.

Si los canales de coordinación entre los departamentos de desarrollo y operaciones no eran adecuados, podían surgir problemas en las condiciones de provisión de los servicios hacia los usuarios clientes y tener un impacto negativo en los niveles de servicio esperados.

DevOps busca reducir los problemas de coordinación entre los departamentos de desarrollo y operaciones, integrando las actividades de desarrollo y operación en un solo flujo.

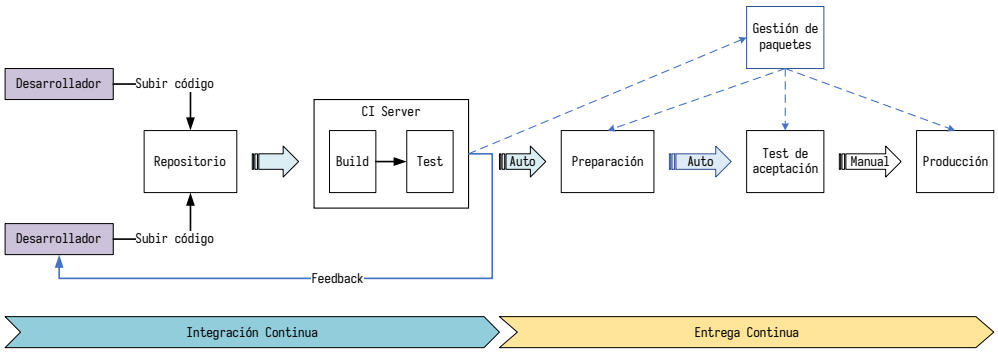
A. Elementos de DevOps. DevOps se funda en los siguientes tres elementos (Kaiser, 2018):

- Personas: los equipos de desarrollo y operaciones que deben crearse alrededor de una aplicación.
- Procesos: integración continua, entrega continua y despliegue continuo.
- Integración continua (Continuous Integration). Práctica de desarrollo de software en el que los miembros de un equipo integran su trabajo con frecuencia en un repositorio de código fuente.
- Entrega continua (Continuous Delivery). Proceso de verificación del código binario resultante de la integración continua desde varias perspectivas para su validación y posterior despliegue en producción, bajo el control y discreción del administrador de lanzamiento.

  El flujo DevOps se describe en el siguiente diagrama tomado de (Krief, 2019):

Figura 12

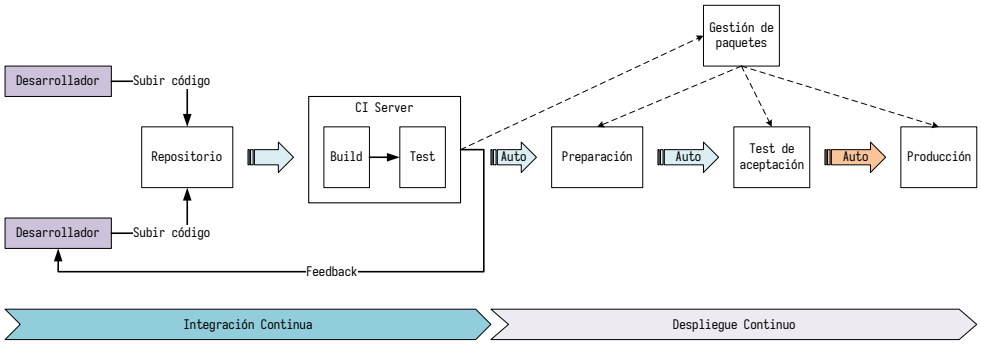
Flujo DevOps con entrega continua



Despliegue continuo (Continuous Deployment). Flujo similar al de entrega continua en el que el despliegue en producción ocurre de forma automática y sin intervención del administrador de lanzamiento. Se describe gráficamente en la siguiente figura tomada de (Krief, 2019):

Figura 13

Flujo DevOps con despliegue continuo



Tecnología: comprende herramientas de automatización que dan soporte a las personas y procesos DevOps. Las herramientas deben proveer soporte a:

- Repositorios de código fuente
- Servicios de hosting
- Orquestadores
- Provisión de entorno y despliegue automático
- Pruebas

2.7. Seguridad en la tecnología de contenedores

El Instituto Nacional de Estándares y Tecnología (NIST) de Estados Unidos, mediante el documento NIST Special Publication 800-190 formula una guía de seguridad de contenedores de aplicación (Souppaya et al., 2017), en la que se describe los riesgos a los que pueden estar expuestos los contenedores de aplicaciones y las contramedidas recomendadas para cada caso. Estas se resumen en la siguiente tabla:

Tabla 4

Resumen de Guía de seguridad de Contenedores NIST SP 800-190

COMPONENTE	RIESGOS	CONTRAMEDIDAS
Imágenes	Vulnerabilidades de las imágenes	Uso de herramientas y procesos: - Integrados al ciclo de vida íntegro de las imágenes. - Que permitan visibilidad de las vulnerabilidades en todas las capas de las imágenes. - Que permitan su imposición mediante políticas.
	Defectos de configuración de imágenes	- Validación de configuración de imágenes. - Monitoreo y reporte continuo del estado de cumplimiento de imágenes. - Imposición de requerimientos de cumplimiento. - Uso de capas base solo de fuentes confiables.
	Malware embebido	- Monitorear imágenes en busca de malware embebido.
	Secretos en texto claro embebidos	- Gestión de secretos fuera de las imágenes, basado en

COMPONENTE	RIESGOS	CONTRAMEDIDAS
		configuración predefinida y controlada por el administrador.
	Uso de imágenes no confiables	- Mantenimiento de un conjunto de imágenes y registros fiables. - Aseguramiento que solo imágenes autorizadas del conjunto se ejecuten.
Registro	Conexiones inseguras a registros	- Uso de canales encriptados
	Imágenes obsoletas en registros	- Limpieza de registros para eliminar imágenes que ya no se debe usar. - Uso de nombres adecuados para las imágenes.
	Restricciones de autenticación y autorización insuficientes	- Autenticación para el acceso a imágenes sensibles.
Orquestador	Acceso administrativo ilimitado	- Usar modelo de acceso de menor privilegio.
	Acceso no autorizado	- Uso de métodos de autenticación
	Trafico entre contenedores pobremente separado	- Uso de redes virtuales para separar el tráfico de red.
	Mezcla de niveles de sensibilidad de carga de trabajo	- Aislamiento de despliegue según niveles de sensibilidad.

COMPONENTE	RIESGOS	CONTRAMEDIDAS
	Confiabilidad del nodo orquestador	- Creación de entornos seguros para las aplicaciones ejecutadas.
Contenedor	Vulnerabilidades en el software runtime	- Monitoreo de vulnerabilidades en el runtime.
	Acceso de red no limitado desde contenedores	- Control de salida de tráfico desde contenedores.
	Configuraciones no seguras del runtime de contenedores	- Automatización del cumplimiento de estándares de configuración del runtime de contenedores.
	Vulnerabilidad de aplicaciones	- Uso de herramientas de prevención y detección de anomalías en tiempo de ejecución.
	Contenedores malintencionados	- Instituir entornos de desarrollo, pruebas y producción separados.
Sistema operativo anfitrión	Superficie de ataque grande	- Usar sistemas operativos específicamente diseñados para ejecutar contenedores.
	Kernel compartido	- Evitar usar un mismo host para servicios en contenedores y servicios que no se ejecutan en contenedores.

COMPONENTE	RIESGOS	CONTRAMEDIDAS
	Vulnerabilidades de componentes del sistema operativo anfitrión	- Verificación y actualización regular de componentes del sistema operativo.
	Derechos de acceso de usuarios impropios	- Control de acceso anómalo al sistema operativo.
	Alteración del sistema de archivos del sistema operativo anfitrión	- Control de políticas de acceso al sistema de archivos en el host.

III. MÉTODO

3.1. Tipo de investigación.

Básica: la presente investigación se llevará a cabo con la finalidad de mejorar el proceso de provisión de servicios web en la RCU – UNSAAC, se utilizarán como base diferentes metodologías existentes, así mismo los aportes en base al conocimiento de la materia, para lograr elaborar una nueva metodología, la cual este acorde a la realidad de la institución y otras similares bajo el mismo sistema. La nueva metodología tendrá por nombre MFDC.

Aplicada: se podrá realizar la aplicación de la nueva metodología MFDC, para que por intermedio del uso de contenedores lograr mejorar la gestión de provisión de servicios web en la RCU – UNSAAC y otras instituciones con similares necesidades.

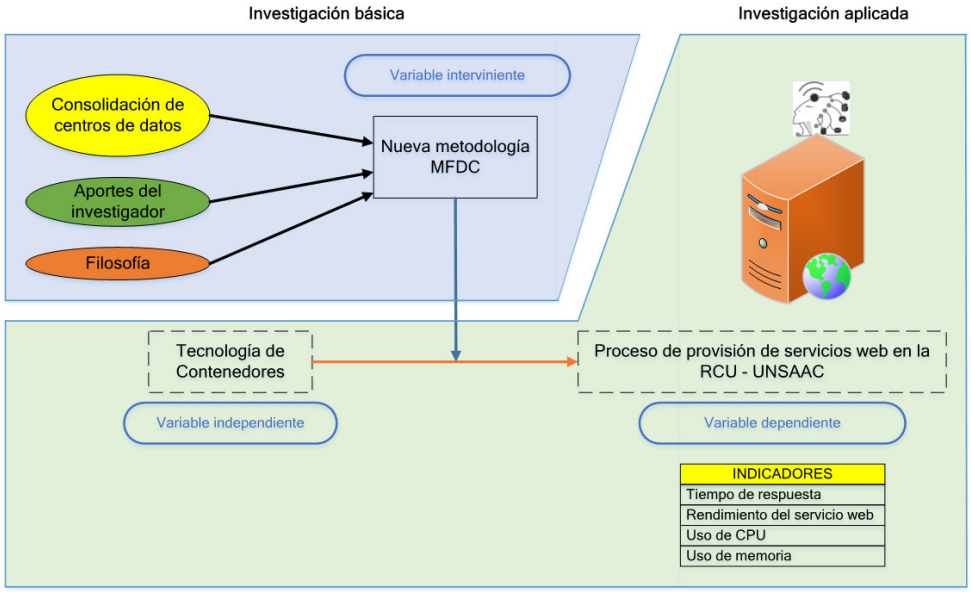
Nivel de investigación

Descriptiva: Descriptiva porque nos permite mostrar la RCU – UNSAAC y el proceso de provisión de servicios web en las condiciones que se encuentra, de igual modo poder encontrar sus variables y así poder medirlas.

Correlacional – causal: porque aplicando la variable independiente, se optimizará la variable dependiente.

Figura 14

Esquema de investigación



3.2. Población y muestra.

UNIDAD MUESTRAL

Proceso de provisión de servicios web

RESTRICCIONES:

- El proceso de provisión de servicios web en la RCU.

UNIVERSO

Todos los procesos de provisión de servicios web de la RCU.

Debido a la viabilidad de determinar la totalidad de procesos de provisión de servicios web, establecemos:

N = Indeterminado

MUESTRA

Considerando que los procesos de provisión de servicios web son similares, para el presente estudio se considera:

n = 60

TIPO DE MUESTREO

Es de tipo aleatorio.

3.3. Operacionalización de variables.

Diseño experimental puro, con pos pruebas y grupos de control.

RGe	X	O1.
RGc	-	O2.

Dónde:

R : Participantes (procesos) elegidos de forma aleatoria.

Ge : Grupo experimental al que se aplicará el estímulo (Tecnología de contenedores).

Gc : Grupo de control al que no se aplicará el estímulo (Tecnología de contenedores).

O1 : Medición de los datos de la variable dependiente antes de la aplicación de la variable independiente.

O2 : Medición de los datos de la variable dependiente después de la aplicación de la variable independiente.

X : Tecnología de contenedores (estímulo o condición experimental).

- : Falta de estímulo o condición experimental.

VARIABLES

Tabla 5

Identificación de variables

VARIABLES	INDICADORES
1. Independiente. Tecnología de contenedores.	Presencia_Ausencia.
2. Dependiente. Proceso de provisión de servicios web en la RCU.	Tiempo de respuesta. Rendimiento del servicio web. Uso de CPU. Uso de Memoria.

INDICADORES

- Conceptualización

Tabla 6

Conceptualización de variable independiente: Tecnología de contenedores

INDICADOR	DEFINICIÓN
Presencia_Ausencia	SI - La tecnología de contenedores fue desplegada. NO - La tecnología de contenedores no fue desplegada.

Tabla 7

Conceptualización de variables dependientes: Mejora del proceso de provisión de servicios web en la RCU

INDICADORES	DEFINICIÓN
Tiempo de respuesta	Mide el tiempo desde que arriba una petición HTTP al servidor hasta que el servidor devuelve una respuesta HTTP.

INDICADORES	DEFINICIÓN
Rendimiento del servicio web	Mide el número de peticiones que puede atender el servidor web por unidad de tiempo.
Uso de CPU	Mide el tiempo de CPU utilizado por una tarea en relación a la suma de dicho tiempo más el tiempo de CPU utilizado para otra tarea distinta.
Uso de Memoria	Mide la cantidad de memoria utilizada por una tarea en relación a la cantidad total de memoria disponible en el sistema.

- Operacionalización

Tabla 8

Operacionalización de variable independiente: Tecnología de contenedores

INDICADOR	ÍNDICES
Presencia_Ausencia	SI/NO

Tabla 9

Operacionalización de variables dependientes: Mejora del proceso de provisión de servicios web en la RCU

DIMENSIÓN	INDICADOR	ÍNDICE	UNIDAD DE MEDIDA	UNIDAD DE OBSERVACIÓN	FÓRMULA
Servicio	Tiempo de respuesta	[0..N]	milisegundos	Formato de observaciones	Tiempo de respuesta HTTP - Tiempo de petición HTTP.
	Rendimiento del servicio web	[0..1]	Número de peticiones HTTP atendidas / segundo	Formato de observaciones	Número de peticiones HTTP atendidas/segundo.
Recursos	Uso de CPU	[0..1]	Adimensional	Formato de observaciones	Tiempo de CPU utilizado por tarea /(Tiempo de CPU utilizado por tarea + Tiempo de CPU utilizado para otra tarea).
	Uso de Memoria	[0..1]	Adimensional	Formato de observaciones	Cantidad de Memoria utilizada por una tarea / cantidad total de memoria disponible en el sistema.

3.4. Instrumentos.

Para el estudio se utilizaron los siguientes instrumentos:

- Recopilación de información documental y digital.
- Entrevistas.
- Formatos de observación.

3.5. Procedimientos.

Se desplegó un sitio web utilizando la metodología propuesta en el presente trabajo de investigación.

Después de desplegados el sitio web, se procede a medir los indicadores definidos para el presente estudio.

3.6. Análisis de datos.

Después de recopilar los resultados de las pruebas, se procede a preparar los mismos para su análisis utilizando software especializado, aplicando las técnicas de la estadística descriptiva.

3.7. Consideraciones éticas

Ninguna especial para el trabajo de investigación.

IV. RESULTADOS

4.1. Metodología propuesta

Hecha la revisión de las metodologías existentes, se llegó al punto en el que estas se evaluaron, compararon y se decidió cómo pueden integrarse estas en la metodología propuesta, de forma que esta metodología integre partes de las metodologías existentes para plantear una nueva que aproveche los beneficios que ofrezcan estas.

Para la propuesta de la metodología materia del presente trabajo de tesis se comparó el flujo de despliegue propuesto por la propia organización creadora de Docker y los procesos CD propuestos como parte de la metodología DevOps.

Por la orientación de los procesos CD, en lo que concierne a las actividades conducentes al despliegue en producción, de aplicaciones y servicios web, se puede establecer que la metodología propuesta por Docker Inc. coincide de forma significativa con los procesos CD de la metodología DevOps.

En cuanto a la propuesta de Docker, se pudo percibir un sesgo hacia el uso de la tecnología, antes que al aprovechamiento de esta para el logro de soluciones TI eficientes, en el supuesto que, de por sí, el uso de la tecnología de contenedores es ya una garantía de eficiencia de las soluciones basadas en esta.

Se puede observar que, tanto CD, como el flujo de Docker, no consideran, de forma explícita, los aspectos de seguridad propuestos por NIST y se centran más bien, en el caso de CD, en los aspectos de pruebas de funcionalidad (test de aceptación) y en el caso de Docker, en la gestión de la tecnología de contenedores Docker.

Se puede resumir que el flujo Docker se centra en la tecnología Docker de contenedores y que los procesos CD, en lo que concierne al proceso de despliegue, se centra en los aspectos mayoritariamente funcionales del proceso CD.

Debe resultar fácil de concluir que las metodologías descritas se complementan, pero, aun así, omiten los aspectos de seguridad sugeridos por NIST.

La metodología que se plantea en el presente trabajo, integra estas tres vertientes en una propuesta unificada. Una que contempla los aspectos funcionales, operacionales y de seguridad en las actividades de despliegue de servicios web mediante contenedores.

En la metodología propuesta, el insumo de entrada es el contenido web funcional, resultado de las fases de desarrollo y pruebas de software previas y listo para su despliegue en los servidores de la organización.

A continuación, se describe la metodología propuesta, partiendo del diagrama BPM que muestra las fases que comprende la metodología y los procesos asociados a cada una de ellas.

A. Descripción de la metodología MFDC. El diagrama BPM de la metodología propuesta comprende las fases de:

- Gestión de imágenes de contenedores. En esta fase, los procesos están orientados a la creación de una imagen de contenedor que se ajuste a las especificaciones brindadas por el Administrador Web. Las especificaciones se complementan con las recomendaciones alcanzadas por el Especialista en Seguridad y por el Especialista en Pruebas.

  Tanto las especificaciones como las recomendaciones se utilizan por el Especialista en Operaciones, como base para la configuración de la imagen del contenedor y su posterior creación.

  La imagen creada es validada por los especialistas de seguridad y pruebas.

Si las pruebas son satisfactorias, se da por concluida esta fase y la imagen creada se utilizará para su almacenamiento en el repositorio de imágenes de contenedores y su posterior despliegue.

- Gestión de repositorio. En esta fase, se selecciona el repositorio de imágenes en donde se almacenará la imagen de contenedor creada, desde donde esta se instanciará en la fase de despliegue.

  Si el repositorio aun no existiese, se debe proceder a su configuración, tomando como base para este fin, las recomendaciones del Especialista en seguridad y el Especialista en pruebas.

  El Especialista en operaciones crea el repositorio, el mismo que debe validarse mediante pruebas por parte de los especialistas correspondientes.

  Una vez que se valida el repositorio, este queda expedito para ser utilizado como almacén de las imágenes de contenedores.

- Gestión de despliegue. En esta fase, se define el entorno de ejecución del contenedor que se instanciará para su despliegue.

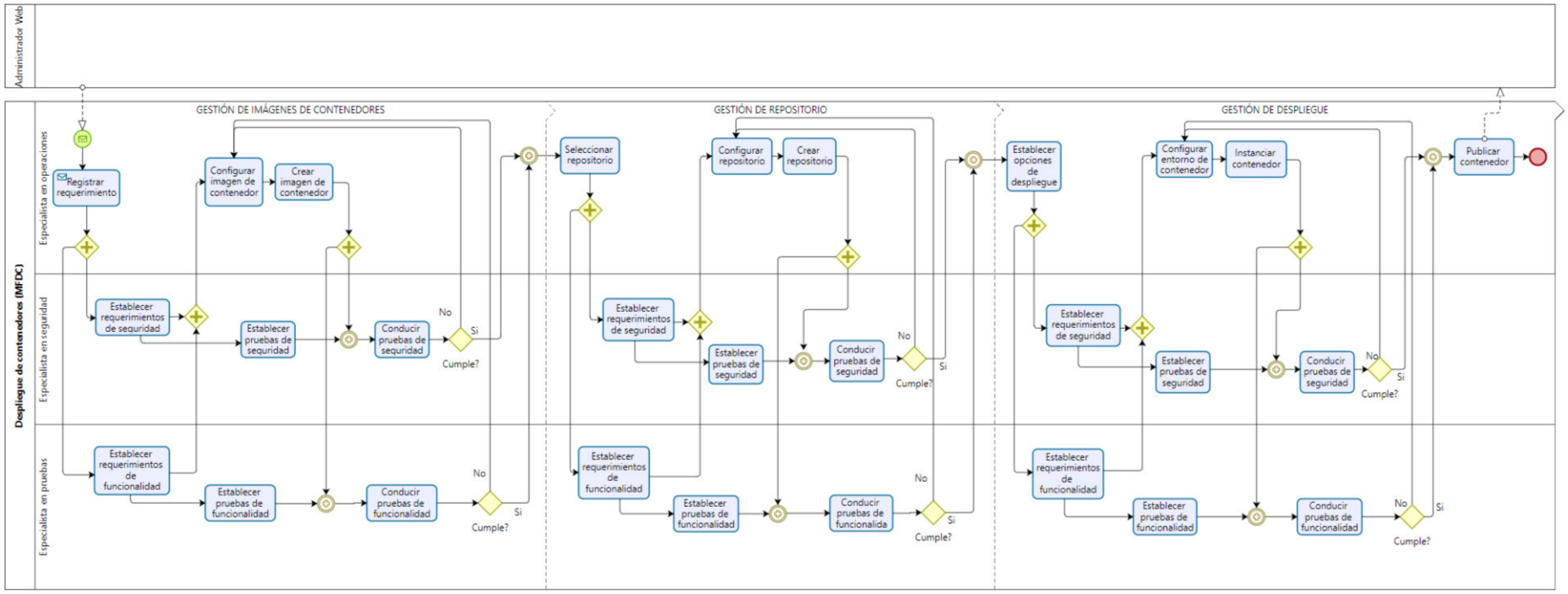
  El entorno de ejecución del contenedor se define a partir de las recomendaciones del Especialista en Seguridad y el Especialista en pruebas.

  El Especialista en operaciones instancia el contenedor y éste es verificado por los especialistas en seguridad y pruebas.

  Si las pruebas son satisfactorias, el Especialista en operaciones despliega el contenedor como un servicio en producción, con lo que el proceso de despliegue del contenedor culmina. Este hecho es comunicado al Administrador web.

Figura 15

Metodología propuesta MFDC



B. Roles y funciones. Como parte de la metodología propuesta, se plantean los siguientes roles:

- Especialista en operaciones.
- Especialista en seguridad.
- Especialista en pruebas.

Las funciones de cada uno de ellos se detallan a continuación:

Especialista en Operaciones

- Funciones
  - Crear las imágenes de contenedores.
  - Crear los repositorios de imágenes de contenedores.
  - Desplegar los contendores.
  - Monitorear los contenedores en operación.
  - Mantener registro de las configuraciones de imágenes de contenedores.
  - Mantener registro de las configuraciones de repositorios de imágenes de contenedores.
  - Mantener registro de operación de contenedores.

Especialista en Pruebas

- Funciones
  - Planificar pruebas de verificación de imágenes de contenedores.
  - Planificar pruebas de verificación de repositorios para imágenes de contenedores.

- Planificar pruebas de verificación de contenedores previos al despliegue.
- Conducir pruebas de verificación de imágenes de contenedores.
- Conducir pruebas de verificación de repositorios para imágenes de contenedores.
- Conducir pruebas de verificación de contenedores previas al despliegue.

Especialista en Seguridad

- Funciones
  - Establecer los requerimientos de seguridad de imágenes de contenedores
  - Validar y verificar los requerimientos de seguridad de imágenes de contenedores.
  - Establecer los requerimientos de seguridad de repositorios de imágenes de contenedores.
  - Validar y verificar los requerimientos de seguridad de imágenes de contenedores.
  - Establecer los requerimientos de seguridad de los contenedores en ejecución.
  - Validar y verificar los requerimientos de seguridad de los contenedores en ejecución.

C. Procesos. Además de los roles propuestos, la metodología plantea las actividades asociadas a cada fase y vincula cada uno de ellas a los roles antes descritos.

Los procesos propuestos, así como las actividades planteadas se describen a continuación:

1) Gestión de imágenes de contenedores. En esta fase, el especialista en operaciones define las características de las imágenes de los contenedores y crea las imágenes correspondientes, considerando las recomendaciones de seguridad y funcionalidad propuestos por los especialistas en seguridad y funcionalidad. Las actividades correspondientes son:

   a) Registrar requerimiento. El administrador web envía el requerimiento de despliegue del contenedor. El especialista en operaciones registra el requerimiento, junto con las especificaciones del servicio de red que debe desplegarse. Las especificaciones de despliegue son derivadas a los especialistas en seguridad y funcionalidad.

   b) Establecer requerimientos de seguridad. El especialista en Seguridad define las especificaciones de seguridad y las alcanza al especialista en operaciones para que sean incluidas en la creación de la imagen del contenedor.

   c) Establecer requerimientos de funcionalidad. El especialista en pruebas establece las especificaciones de funcionalidad para la creación de la imagen del contenedor y las alcanza al especialista en operaciones para que sean incluidas en la creación de la imagen del contenedor.

   d) Configurar imagen del contenedor. El especialista en operaciones crea los archivos de configuración del contenedor.

   e) Crear imagen del contenedor. El especialista en operaciones crea la imagen del contenedor a partir de los archivos de configuración elaborados en la actividad anterior.

   f) Establecer pruebas de seguridad. El especialista en seguridad diseña las pruebas de seguridad para verificar y validar la imagen de contenedor creada.

   g) Establecer pruebas de funcionalidad. El especialista en pruebas diseña las pruebas de funcionalidad para verificar la imagen de contenedor creada.

h) Conducir pruebas de seguridad. El especialista en seguridad conduce las pruebas de seguridad de la imagen de contenedor creada. Si las pruebas son satisfactorias, es decir cumplen con las especificaciones de seguridad, se culmina con la fase de gestión de imágenes de contenedores, en otro caso, se alcanzan las observaciones al especialista en operaciones para subsanar las deficiencias detectadas. En este último caso, se repiten las actividades desde la actividad Configurar imagen del contenedor.

i) Conducir pruebas de funcionalidad. El especialista en pruebas conduce pruebas de funcionalidad de la imagen de contenedor creada. Si las pruebas son satisfactorias, se culmina con la fase de gestión de imágenes de contenedores, en otro caso, se alcanzan las observaciones al especialista en operaciones para subsanar las deficiencias detectadas. En este último caso, se repiten las actividades desde la actividad Configurar imagen del contenedor.

2) Gestión de repositorios. En esta fase, se crea el repositorio de imágenes de contenedores. El repositorio se crea tomando en consideración las recomendaciones de los especialistas en seguridad y funcionalidad. Una vez creado el repositorio, se realizan pruebas de seguridad y funcionalidad para validar la plena operatividad del repositorio.

La tarea de creación del repositorio puede realizarse por única vez, puesto que el repositorio podrá ser utilizado para almacenar imágenes de contenedores adicionales.

Las actividades que comprende esta fase son:

a) Seleccionar repositorio. El especialista en operaciones selecciona el repositorio de imágenes de contenedor. Las posibles opciones son: utilizar un repositorio público o crear un repositorio de imágenes de contenedores local propio.

b) Establecer requerimientos de seguridad. El especialista en seguridad define los requerimientos de seguridad que debe cumplir el repositorio de imágenes de contenedores.

c) Establecer requerimientos de funcionalidad. El especialista en pruebas define los requerimientos de funcionalidad del repositorio de imágenes de contendores.

d) Configurar repositorio. El especialista en operaciones crea los archivos de configuración del repositorio de imágenes de contenedores.

e) Crear repositorio. El especialista en operaciones crea el repositorio de imágenes de contenedores.

f) Establecer pruebas de seguridad. El especialista en seguridad diseña las pruebas de seguridad para la validación del repositorio de imágenes de contenedores.

g) Establecer pruebas de funcionalidad. El especialista en pruebas diseña las pruebas de funcionalidad para la validación del repositorio de imágenes de contenedores.

h) Conducir prueba de seguridad. El especialista en seguridad verifica y valida las pruebas se seguridad del repositorio de imágenes de contenedores. Si los resultados de las pruebas no satisfacen los requerimientos establecidos, se comunica al especialista en operaciones para realice las actividades correctivas. Si las pruebas son exitosas, la fase de gestión de repositorio culmina en lo que respecta a seguridad.

i) Conducir pruebas de funcionalidad. El especialista en pruebas conduce pruebas de funcionalidad. Si los resultados de las pruebas no satisfacen los requerimientos establecidos, se comunica al especialista en operaciones para realice las actividades correctivas. Si las pruebas son exitosas, la fase de gestión de repositorio culmina en lo que respecta a funcionalidad.

3) Gestión de despliegue. En esta fase, el especialista en operaciones despliega el contenedor, definiendo los parámetros de operación del contenedor tomando en consideración las recomendaciones de los especialistas en seguridad y funcionalidad.

Si la funcionalidad del contenedor cumple con las especificaciones correspondientes, este es puesto en producción.

Las actividades correspondientes a esta fase son:

a. Establecer opciones de despliegue. El especialista en operaciones define los parámetros de despliegue del contenedor.

b. Establecer requerimientos de seguridad. El especialista en seguridad define los requerimientos de seguridad para el despliegue del contenedor.

c. Establecer requerimientos de funcionalidad. El especialista en pruebas define los requerimientos de funcionalidad para el despliegue del contenedor.

d. Configurar entorno del contenedor. El especialista en operaciones crea los archivos de configuración para el despliegue del contenedor.

e. Instanciar contenedor. El especialista en operaciones instancia la imagen del contenedor.

f. Establecer pruebas de seguridad. El especialista en seguridad diseña las pruebas de seguridad para el despliegue del contenedor.

g. Establecer pruebas de funcionalidad. E especialista en pruebas diseña las pruebas de funcionalidad para el despliegue del contenedor.

h. Conducir prueba de seguridad. El especialista en seguridad conduce las pruebas de seguridad para el contenedor desplegado. Si las pruebas no satisfacen los requerimientos de seguridad, se alcanza al especialista en operaciones los resultados de las pruebas para su corrección. En otro caso, se aprueba los parámetros de seguridad para el despliegue del contenedor.

i. Conducir pruebas de funcionalidad. El especialista en pruebas conduce las pruebas de funcionalidad para el despliegue del contenedor. Si las pruebas no satisfacen los requerimientos de funcionalidad, se alcanza al especialista en operaciones los resultados de las pruebas para su corrección. En otro caso, se aprueba los parámetros de funcionalidad para el despliegue del contenedor.

j. Publicar contenedor. El contenedor es puesto en operación, hecho que se comunica al administrador web con lo que se culmina el despliegue del contenedor.

D. Formatos de documentación de la metodología MFDC. Para la documentación de las actividades en la metodología MFDC se utilizarán formatos creados como parte de la propuesta metodológica a partir de las recomendaciones NIST SP 800-190.

Para la denominación de los formatos, se utiliza el siguiente patrón:

<Tipo><Actividad><Dimensión>

En donde:

<Tipo> puede ser:

F – Formato de configuración.

C – Lista de comprobación (Checklist) de validación de pruebas.

<Actividad> puede ser:

I – Gestión de imágenes de contenedores.

R – Gestión de repositorio de imágenes de contenedores.

D – Gestión de despliegue de contenedores.

<Dimensión> puede ser:

S – Seguridad.

P – Funcionalidad.

Por ejemplo:

FIS – Formato de gestión de imágenes – Seguridad.

CIS – Lista de comprobación de seguridad de imágenes de contenedores.

Los formatos forman parte de los anexos del presente documento.

E. Fundamentos de la metodología MFDC. Normas y estándares:

La metodología propuesta se basa en los siguientes documentos:

- Metodología DevOps.
- Norma NIST SP 800-190.
- Metodología de despliegue de contenedores de Docker Inc.

Filosofía

La tecnología Docker se sustenta en la promesa de construir una vez, desplegar donde sea.

Tomando como principio esta promesa, la metodología propuesta plantea como principio filosófico: construir una vez, desplegar donde sea aplicando un enfoque de seguridad proactivo.

Métricas

Toda metodología debe dar soporte a métricas que permitan su evaluación y validación en términos cualitativos o cuantitativos.

El caso de la MFDC, las métricas que se plantean son:

- Formalidad de los procesos de despliegue de servicios de red – evidenciable por la secuencia de actividades plenamente establecidas.

- Trazabilidad – evidenciable por la documentación generada a partir del uso de la metodología.

La metodología propuesta centra su propósito en el proceso de despliegue de contenedores, por lo que en el proceso de adaptación de las recomendaciones de NIST y DevOps, los aspectos relacionados a la gestión del sistema operativo se omiten por considerarse una actividad que es parte del proceso de gestión de seguridad de la plataforma e independiente del proceso de despliegue de contenedores.

Las recomendaciones de NIST se seleccionaron y agruparon de acuerdo a su nivel de criticidad y las fases que se proponen en la metodología. El listado de las mismas se muestra a continuación:

Tabla 10

Controles de seguridad MFDC

FASE	RIESGOS	ID	CONTROL
Gestión de imágenes de contenedores.	Vulnerabilidades de las imágenes	S1	Verificar vulnerabilidades en las imágenes
	Defectos de configuración de imágenes	S2	Validar configuración de imágenes.
		S3	Usar capas base solo de fuentes confiables.
	Malware embebido	S4	Monitorear imágenes en busca de malware embebido
	Secretos en texto claro embebidos	S5	Verificar presencia de secretos en imágenes de contenedores
	Uso de imágenes no confiables	S6	Mantener conjunto de imágenes fiables.
		S7	Restringir ejecución de imágenes no autorizadas
Gestión de repositorio.	Conexiones inseguras a registros	S8	Utilizar canales encriptados
	Imágenes obsoletas en registros	S9	Limpiar registros para eliminar imágenes que ya no se debe usar
		S10	Usar nombres adecuados para las imágenes.
	Restricciones de autenticación y autorización	S11	Autenticar para el acceso a imágenes sensibles.
Gestión de despliegue.	Acceso administrativo ilimitado	S12	Usar modelo de menor privilegio.
	Confiabilidad del nodo orquestador.	S13	Crear entorno seguro para las aplicaciones ejecutadas.
	Contenedores malintencionados	S14	Instituir entornos de desarrollo, pruebas y producción separados.

Las pruebas de funcionalidad se han establecido con el propósito de establecer capacidades funcionales mínimas para cada una de las fases de la metodología propuesta.

La lista con los requerimientos de funcionalidad para cada fase de MFDC se documenta en la siguiente tabla:

Tabla 11

Controles de funcionalidad MFDC

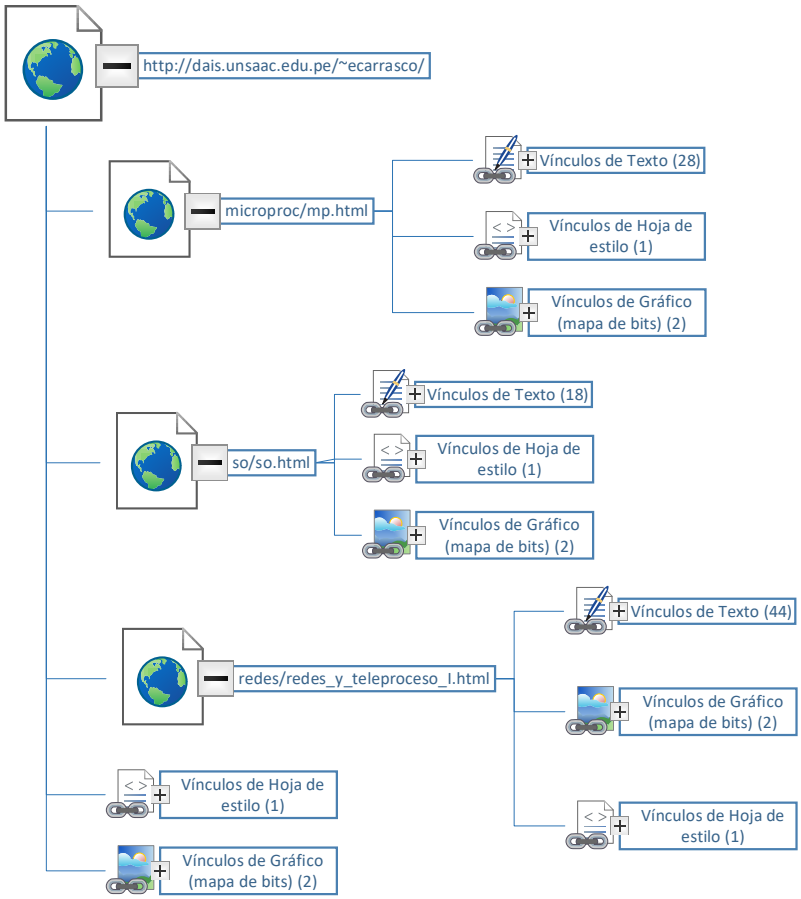
FASE	FUNCIONALIDAD	ID	CONTROL
Gestión de imágenes de contenedores.	Dimensionamiento de servicio	F1	Establecer capacidad de CPU para el servicio
		F2	Establecer capacidad de memoria principal para el servicio
	Acceso al servicio.	F3	Verificar acceso a servicio.
Gestión de repositorio.	Disponibilidad del repositorio	F4	Verificar disponibilidad del acceso al repositorio de imágenes.
	Almacenamiento de imágenes en el repositorio	F5	Verificar la capacidad de almacenar imágenes de contenedores en el repositorio.
	Descarga de imágenes del repositorio.	F6	Verificar la capacidad de descargar imágenes del repositorio de contenedores.
Gestión de despliegue.	Acceso al servicio	F7	Verificar la capacidad de acceder al servicio desplegado
	Gestion del servicio	F8	Verificar la capacidad de controlar el contenedor desplegado
	Acceso a bitácora	F9	Verificar la accesibilidad a bitácoras de monitoreo de servicios.

4.2. Aplicación de la metodología en un caso real

Con el objetivo de verificar la metodología MFDC propuesta, se procedió a aplicar la misma en un ejercicio real, utilizando como insumo de entrada un sitio web diseñado para el propósito del trabajo de investigación y cuyo mapa se documenta en la siguiente figura:

Figura 16

Mapa de sitio web



La secuencia de actividades que se ejecutaron para la aplicación y validación de MFDC son:

1. Aplicación de la metodología propuesta para el despliegue del sitio web diseñado utilizando la tecnología de contenedores. Se documenta en el anexo C.
2. Despliegue del sitio web en la plataforma que actualmente es utilizada por la institución donde se realizaron las pruebas. Se documenta en el anexo D.
3. Pruebas comparativas de acuerdo a los criterios de evaluación del presente estudio.
4. Análisis de resultados.

MFDC inicia con el envío del requerimiento de despliegue del sitio web por parte del Administrador web.

El contenido del sitio web es enviado por el Administrador Web junto con la ficha de especificaciones ES al Especialista en Operaciones. Este punto marca el inicio de la MFDC.

Se procede con la primera fase de la metodología propuesta.

A. Gestión de imágenes de contenedores. Registrar requerimiento

Esta actividad marca el inicio de la metodología propuesta. Se registra el requerimiento, del Administrador Web y se recibe el código y contenido del sitio web que se debe desplegar.

Las especificaciones del sitio web se registran en el formato propuesto:

Figura 17

Formato de especificaciones de sitio web

ES	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO	Despliegue de sitio web Portal Docente	
ACTIVIDAD	ESPECIFICACIONES DE SITIO WEB	
RESPONSABLE	Especialista en operaciones	
FECHA	14/11/2022	
ESPECIFICACIONES		
1	Nombre de archivo del sitio web	public_html
2	Dominio del sitio web	http://dais.unsaac.edu.pe/~ecarrasco/
3	Tecnología	HTML, CSS, PDF
4	Población objetivo	General
5	Disponibilidad	No crítico
6	Restricciones de modificación	Inmutable
7	Restricciones de acceso	Ninguna

La ficha de especificaciones del sitio se remite a los especialistas en seguridad y pruebas.

El Especialista en Seguridad completa el formato FIS con las tareas pertinentes para el tipo de proyecto y las envía al especialista en operaciones. Las tareas pueden incluirse en su

totalidad o podrían incluir solo las que el experto considere necesarias para el proyecto. Para el caso del proyecto de aplicación de MFDC, se incluyen todas las tareas de seguridad.

Figura 18

Formato de recomendaciones de seguridad de imágenes

FIS	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO	Despliegue de sitio web Portal Docente	
ACTIVIDAD	RECOMENDACIONES DE SEGURIDAD DE IMÁGENES	
RESPONSABLE	Especialista en seguridad	
FECHA	16/11/2022	
TAREAS		Observación
1	Utilizar imagen base oficial o de fuente reconocida	S3
2	Utilizar archivos de origen fiable para la creación de la imagen del contenedor	S1, S2
3	Controlar los componentes que se incluyen en el contenedor	S4, S5
4	Eliminar toda imagen de contenedor que no sea destinada a producción	S6, S7

Por su parte, el Especialista en Pruebas envía el formato FIP con las tareas de control de funcionalidad sugeridas. El formato se muestra a continuación:

Figura 19

Formato de recomendaciones de funcionalidad de imágenes

FIP	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO	Despliegue de sitio web Portal Docente	
ACTIVIDAD	RECOMENDACIONES DE FUNCIONALIDAD DE IMÁGENES	
RESPONSABLE	Especialista en pruebas	
FECHA	16/11/2022	
TAREAS		Observación
1	Establecer recursos hardware requeridos por la aplicación	F1, F2
2	Verificar acceso al servicio	F3

Estas fichas son utilizadas por el Especialista en Operaciones para la creación de la imagen de contenedor para el servicio que se desplegará.

Las recomendaciones son atendidas por el Especialista en Operaciones de acuerdo al siguiente detalle:

S3: Usar capas base solo de fuentes fiables.

Acción: Para el caso de aplicación, se utiliza la capa base httpd del repositorio oficial de Apache (https://hub.docker.com/_/httpd).

S1: Verificar vulnerabilidad en las imágenes.

Acción: Se procede a verificar el contenido del sitio web por posibles archivos que constituyan una amenaza a la seguridad del sitio. Esta tarea se completa utilizando el software antivirus disponible en la plataforma Linux: Clam-AV.

Figura 20

Verificación de vulnerabilidad de contenido web

```
root@contenedorHP:/home/ecp/webInfo# clamscan -r www/
/home/ecp/webInfo/www/mm_arrow.gif: OK
/home/ecp/webInfo/www/mm_travel_photo.jpg: OK
/home/ecp/webInfo/www/so/so.html: OK
/home/ecp/webInfo/www/so/guias/guia09-tuberias/tuberias.pdf: OK
/home/ecp/webInfo/www/so/guias/guia09-tuberias/_notes/dwsync.xml: OK
/home/ecp/webInfo/www/so/guias/guia07-archivosProyectados/archivosProyectados.pdf: OK
/home/ecp/webInfo/www/so/guias/guia07-archivosProyectados/_notes/dwsync.xml: OK
/home/ecp/webInfo/www/so/guias/guia05-hilos/gestionHilos.pdf: OK
```

S2: Validar configuración de imágenes.

Acción: la configuración de imagen se define en el archivo Dockerfile. En este archivo se define las características de la imagen como, por ejemplo:

Origen de la imagen base, que se establece al repositorio oficial del servidor web que se utilizara: Apache.

Así mismo, se puede incluir comandos que se ejecutaran al momento de crear la imagen, que para este caso comprende el comando de copiar para copiar el sitio web del archivo local a la imagen que se construye.

Seguidamente se muestra el contenido del archivo Dockerfile:

```
#Imagen de origen
FROM hhtpd:2.4.55

#copiar sitio web a contenedor
COPY ./www /usr/local/apache2/htdocs
```

S4: Monitorear imágenes en busca de malware embebido.

Acción: El uso de la imagen base oficial y el escaneo de los archivos que se desplegarán en el servidor web mitigan el riesgo de malware embebido en imagen de contenedor.

S5: Verificar presencia de secretos en imagen de contenedores.

Acción: La creación local de los archivos de configuración para la creación de la imagen del contendor muestran que no se incluye información que constituya secretos que puedan ser utilizados al desplegarse contenedores basados en la imagen construida.

S6: Mantener conjunto de imágenes fiables.

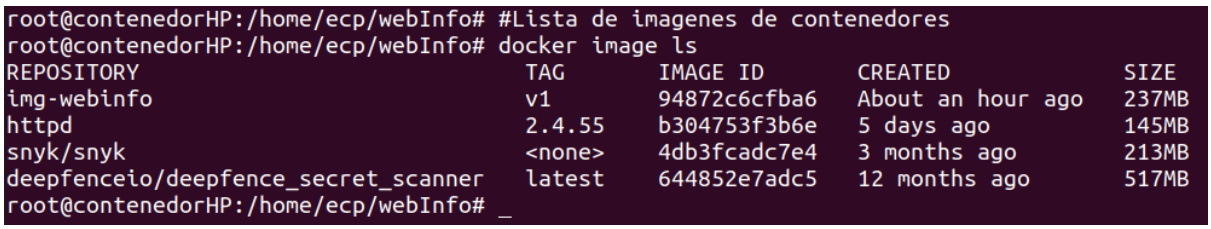
S7: Restringir ejecución de imágenes no autorizadas.

Acción: Para mitigar estas posibles amenazas, se mantiene la lista de imágenes de contenedores eliminando cualquier imagen que no sea parte del proyecto actual. Se verifica este hecho utilizando el comando:

`docker image ls`.

Figura 21

Lista de imágenes de contenedores



F1: Establecer capacidad de CPU para el servicio.

Acción: tomando como criterio el tamaño del sitio web (87MB, distribuidos en 181 archivos) se estima suficiente utilizar un CPU.

F2: Establecer capacidad de memoria principal para el servicio.

Acción: con las consideraciones expuestas para F1, se asigna 4GB de memoria RAM.

F3: Verificar acceso al servicio.

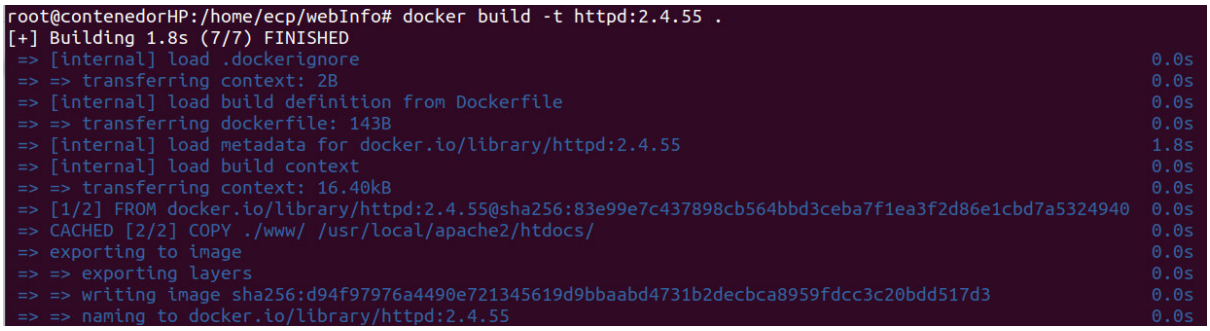
Acción: se crea el contenedor de manera que se verifique la accesibilidad del servicio y se elimine automáticamente el mismo siguiendo los siguientes pasos:

- Construir imagen mediante el comando:

  `docker build -t httpd:2.4.55 .`:

Figura 22

Creación de imágen de contenedor

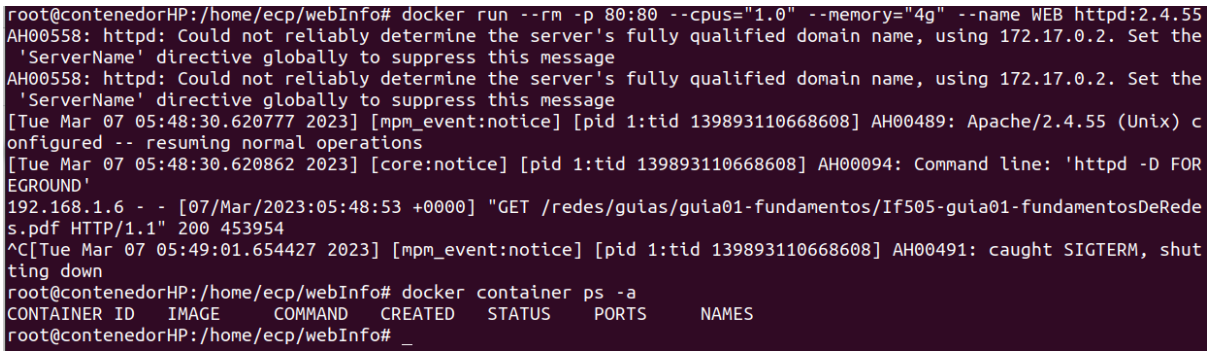


- Se crea un contenedor utilizando la imagen creada de forma que se elimine el mismo luego de hechas las pruebas de acceso mediante el comando: `docker run --rm -p 80:80 --cpus=”1.0” --memory=”4g” --name WEB httpd:2.4.55`.

Desde un navegador web se accede al sitio web, hecho que se verifica por los mensajes mostrados en el terminal.

Figura 23

Verificación de acceso al servicio



De esta manera, se verifica la accesibilidad al servicio desplegado mediante la imagen de contenedor creada.

Concluida la construcción de la imagen de contenedor para el despliegue del servicio web, esta es puesta a disposición de los especialistas en seguridad y pruebas para su verificación y validación, siguiendo el flujo de trabajo de MFDC.

PRUEBAS DE SEGURIDAD DE IMAGEN DE CONTENEDOR

Para las pruebas de seguridad se utiliza el formato CIS:

Vulnerabilidad de imagen:

Para verificar vulnerabilidades en la imagen de contenedor, utilizamos la herramienta dockle (https://github.com/goodwithtech/dockle).

Dockle además, de detectar vulnerabilidades en la imagen, ayuda en la creación basada en buenas prácticas, del archivo Dockerfile:

Figura 24

Escaneo de vulnerabilidades en imágen de contenedor

```
root@contenedorHP:/home/ecp/webInfo# dockle httpd:2.4.55
INFO    - CIS-DI-0005: Enable Content trust for Docker
        * export DOCKER_CONTENT_TRUST=1 before docker pull/build
INFO    - CIS-DI-0006: Add HEALTHCHECK instruction to the container image
        * not found HEALTHCHECK statement
INFO    - CIS-DI-0008: Confirm safety of setuid/setgid files
        * setuid file: urwxr-xr-x usr/bin/passwd
        * setuid file: urwxr-xr-x usr/bin/chfn
        * setuid file: urwxr-xr-x usr/bin/gpasswd
        * setgid file: grwxr-xr-x sbin/unix_chkpwd
        * setuid file: urwxr-xr-x bin/su
        * setgid file: grwxr-xr-x usr/bin/expiry
        * setgid file: grwxr-xr-x usr/bin/wall
        * setuid file: urwxr-xr-x bin/umount
        * setgid file: grwxr-xr-x usr/bin/chage
        * setuid file: urwxr-xr-x usr/local/apache2/bin/suexec
        * setuid file: urwxr-xr-x usr/bin/chsh
        * setuid file: urwxr-xr-x usr/bin/newgrp
        * setuid file: urwxr-xr-x bin/mount
```

Se verifica que la imagen no presenta vulnerabilidades.

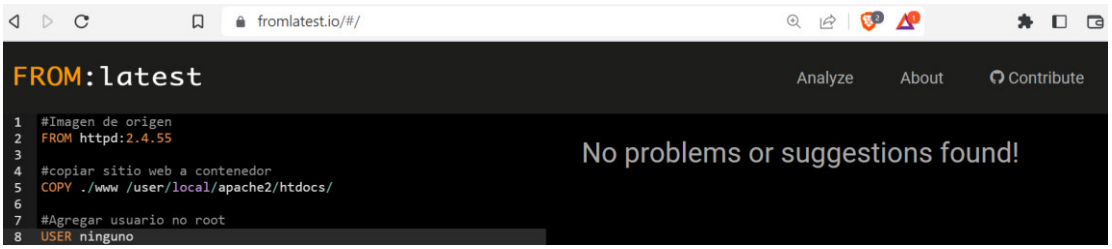
Configuración de imagen válida:

Se utiliza la aplicación Dockerfilelint, disponible en línea en el enlace https://github.com/replicatedhq/dockerfilelint, para evaluar la configuración del archivo Dockerfile, donde se especifican las características de la imagen que se construirá.

Como puede verse en la figura, el archivo no presenta errores.

Figura 25

Validación de archivo Dockerfile



Capa base de fuente fiable:

Se verifica el origen de la imagen base mediante la verificación del archivo Dockerfile, donde se muestra que la imagen proviene del repositorio oficial de Apache.

Figura 26

Listado de imágenes de contenedor en repositorio local

```
root@contenedorHP:/home/ecp# #Listar imagenes base http
root@contenedorHP:/home/ecp# docker search httpd
NAME                              DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
httpd                             The Apache HTTP Server Project                  4356      [OK]
clearlinux/httpd                  httpd HyperText Transfer Protocol (HTTP) ser…   2
paketobuildpacks/httpd                                                            0
avenga/httpd-static                                                               0
betterweb/httpd                                                                   0
centos/httpd-24-centos7           Platform for running Apache httpd 2.4 or bui…   45
manageiq/httpd                    Container with httpd, built on CentOS for Ma…   1                    [OK]
centos/httpd-24-centos8                                                           1
```

Malware embebido:

Se descarta la presencia de malware embebido como resultado de la evaluación de la imagen con el comando dockle ya mostrado.

Secretos en imágenes de contenedores:

Se descarta la presencia de secretos en el contenedor, mediante la verificación directa del archivo Dockerfile y también, utilizando una herramienta específicamente diseñada para tal propósito: la herramienta secretscanner (https://github.com/deepfence/SecretScanner). El resultado de la prueba se muestra a continuación:

Figura 27

Escaneo de secretos en imagen de contenedor

```
Initializing....
Scanning image httpd:2.4.55 for secrets...
[36mINFO[0m[2023-03-07 16:03:13] trying to connect to endpoint 'unix:///var/run/docker.sock' with timeout '10s'
[36mINFO[0m[2023-03-07 16:03:13] connected successfully using endpoint: unix:///var/run/docker.sock
[36mINFO[0m[2023-03-07 16:03:13] trying to connect to endpoint 'unix:///run/containerd/containerd.sock' with timeout '10s'
[33mWARN[0m[2023-03-07 16:03:23] could not connect to endpoint 'unix:///run/containerd/containerd.sock': context deadline exceeded
[36mINFO[0m[2023-03-07 16:03:23] trying to connect to endpoint 'unix:///run/k3s/containerd/containerd.sock' with timeout '10s'
[33mWARN[0m[2023-03-07 16:03:33] could not connect to endpoint 'unix:///run/k3s/containerd/containerd.sock': context deadline exceeded
[36mINFO[0m[2023-03-07 16:03:33] container runtime detected: docker
Scanning image /tmp/Deepfence/SecretScanning/df_httpd2455/save-output.tar for secrets...

.....

  }{
  "Timestamp": "2023-03-07 16:03:37.554822514 +00:00",
  "Image Name": "httpd:2.4.55",
  "Image ID": "e4ace427a241144413a651903ee7db0ee214abd2cc9b64e65ffaeb79a0741d03",
  "Secrets": [

  ]
}
```

Conjunto de imágenes fiables:

Se verifica mediante el comando:

`docker images`,

que muestra las imágenes creadas en el equipo local.

Presencia de imágenes no autorizadas.

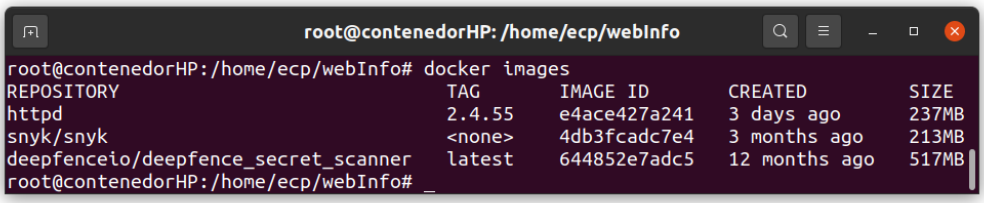
Se verifica listando las imágenes presentes en el equipo local utilizando el comando:

`docker images`.

Los dos requerimientos previos se muestran en la siguiente imagen:

Figura 28

Listado de imágenes de contenedor autorizadas



El resultado de la verificación, se resume en el formato CIS que se muestra a continuación:

Figura 29

Resultado de evaluación de seguridad de imágenes

CIS	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES		
PROYECTO	Despliegue de sitio web Portal Docente		
ACTIVIDAD	LISTA DE COMPROBACION DE SEGURIDAD DE IMÁGENES		
RESPONSABLE	Especialista en seguridad		
FECHA	21/11/2022		
TAREAS		Resultado	Observación
1	Imagen vulnerable	NO	
2	Configuración de imagen válida	SI	
3	Capa base de fuente fiable	SI	
4	Malware embebido	NO	
5	Secretos en imágenes de contenedores	NO	
6	Conjunto de imágenes fiables	SI	
7	Presencia de imágenes no autorizadas	NO	

PRUEBAS DE FUNCIONALIDAD DE IMAGEN DE CONTENEDOR

El Especialista en Pruebas conduce las pruebas de funcionalidad y completa el formato CIP.

Capacidad de CPU establecida:

Capacidad de memoria principal establecida:

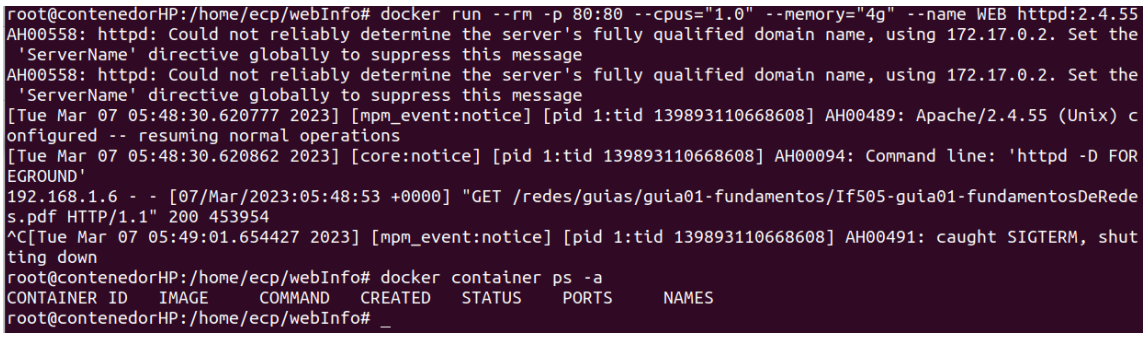
Acceso al servicio:

Se verifica el cumplimiento de estos requerimientos repitiendo la ejecución del comando

`docker run --rm -p 80:80 --cpus=”1.0” --memory=”4g” --name WEB httpd:2.4.55`

Figura 30

Prueba de asignación de recursos a contenedor



El resultado de las pruebas se registra en el formato CIP

Figura 31

Resultado de evaluación de funcionalidad de imágenes

CIP	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES		
PROYECTO	Despliegue de sitio web Portal Docente		
ACTIVIDAD	LISTA DE COMPROBACION DE FUNCIONALIDAD DE IMÁGENES		
RESPONSABLE	Especialista en pruebas		
FECHA	21/11/2022		
TAREAS		Resultado	Observación
1	Capacidad de CPU establecida	SI	
2	Capacidad de memoria principal establecida	SI	
3	Acceso al servicio	SI	

Puesto que las pruebas han sido satisfactorias, la fase de Gestión de Imágenes de Contenedores concluye con un entregable consistente de la imagen de contenedor construida y los archivos de configuración de la imagen. Se procede a la siguiente fase: la fase de Gestión de Repositorio.

B. Gestión de repositorio. Para la fase de Gestión de Repositorio, el insumo de entrada es la imagen de contenedor entregada en la fase de Gestión de Imágenes de Contenedores.

La segunda fase de MFDC inicia con la selección del repositorio de imágenes de contenedores que se utilizará para el almacenamiento de la imagen de contenedor creada en la fase previa.

Para el caso del presente proyecto de aplicación de MFDC, se opta por utilizar Docker Hub como repositorio de las imágenes creadas.

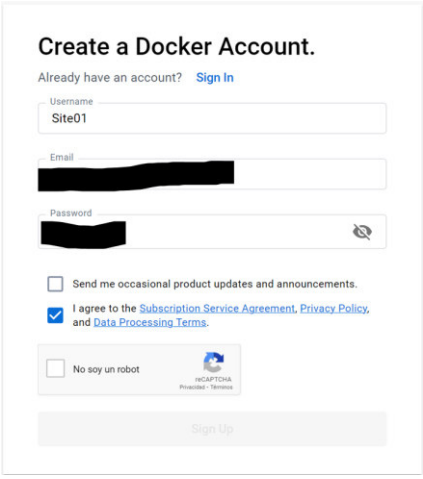
Docker Hub ofrece un servicio gratuito para el almacenamiento de imágenes de contenedores para casos de prueba. La elección de este servicio libera a la organización de la implementación de una plataforma hardware y software para la gestión de imágenes de contenedores. En caso de adoptarse la tecnología de contenedores, puede evaluarse la viabilidad de esta alternativa, considerando factores como la disponibilidad del servicio, la tolerancia a fallos, redundancia y las tareas de gestión operacional y administrativa, así como los costos asociados a las capacidades indicadas.

Para acceder al servicio de Docker Hub, se crea una cuenta a nombre del usuario Site01

Usuario: Site01

Figura 32

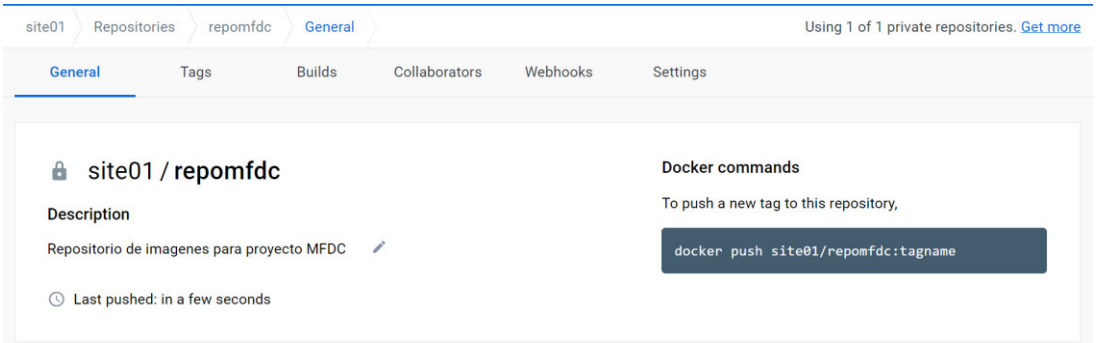
Creación de cuenta en Docker



A continuación, se crea el repositorio repomfdc:

Figura 33

Creación de repositorio



Quedando listo para su uso.

Una vez seleccionada el repositorio, el Especialista en Operaciones solicita las recomendaciones de seguridad y pruebas a los especialistas correspondientes.

El Especialista en Seguridad formula sus recomendaciones utilizando el formato FRS

Figura 34

Formato de recomendaciones de seguridad de repositorio

FRS	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO	Despliegue de sitio web Portal Docente	
ACTIVIDAD	RECOMENDACIONES DE SEGURIDAD DE REPOSITORIO	
RESPONSABLE	Especialista en seguridad	
FECHA	28/11/2022	
TAREAS		Observación
1	Verificar seguridad de conexiones a registros	S8
2	Verificar vigencia de imágenes en registros	S9
3	Verificar denominación de imágenes en registros	S10
4	Verificar autenticación y autorización para las conexiones al registro	S11

El Especialista en Pruebas envía por su parte, sus recomendaciones utilizando el formato FRP.

Figura 35

Formato de recomendaciones de funcionalidad de repositorio

FRP	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO	Despliegue de sitio web Portal Docente	
ACTIVIDAD	RECOMENDACIONES DE FUNCIONALIDAD DE REPOSITORIO	
RESPONSABLE	Especialista en pruebas	
FECHA	28/11/2022	
TAREAS		Observaciones
1	Verificar disponibilidad de acceso a repositorio de imágenes	F4
2	Verificar la capacidad de almacenar imágenes de contenedores en el repositorio	F5
3	Verificar la capacidad de descargar imágenes del repositorio de contenedores	F6

Estos formatos son utilizados por el Especialista en Operaciones, que despliega la imagen de contenedor en el repositorio seleccionado, considerando las recomendaciones como se documenta a continuación:

S8: Utilizar canales encriptados

Acción: el acceso al repositorio de imágenes es mediante cuenta validada y el sitio utiliza conexión segura: https://hub.docker.com/repository/docker/site01/repomfdc/general.

S9: Limpiar registros para eliminar imágenes que ya no se debe usar.

Acción: Se verifica que el repositorio contenga solo la imagen asociada al presente proyecto.

S10: Usar nombres adecuados para las imágenes.

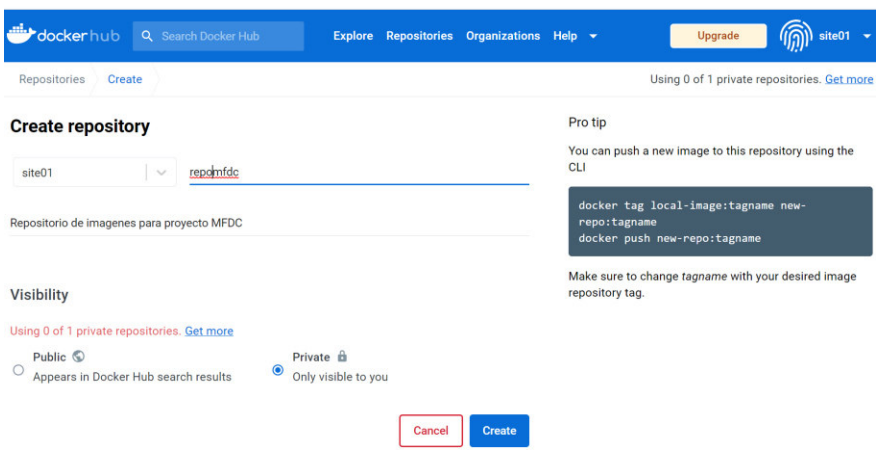
Acción: Se asigna un nombre consistente con el propósito de la imagen: repofmdc:webinfo.

S11: Autenticar para el acceso a las imágenes sensibles:

Acción: El repositorio se crea como repositorio privado. Asegurando que solo sea visible a usuarios autorizados.

Figura 36

Creación de repositorio privado

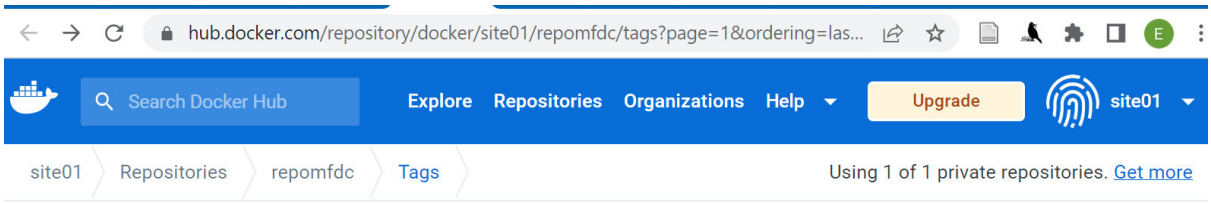


F4: Verificar disponibilidad del acceso al repositorio de imágenes.

Acción: Se ingresa al repositorio de imágenes desde un navegador web, utilizando los datos de la cuanta ya establecidos. El resultado es satisfactorio, demostrándose el acceso al mismo, como se evidencia en la siguiente imagen:

Figura 37

Disponibilidad de acceso a repositorio de imágenes



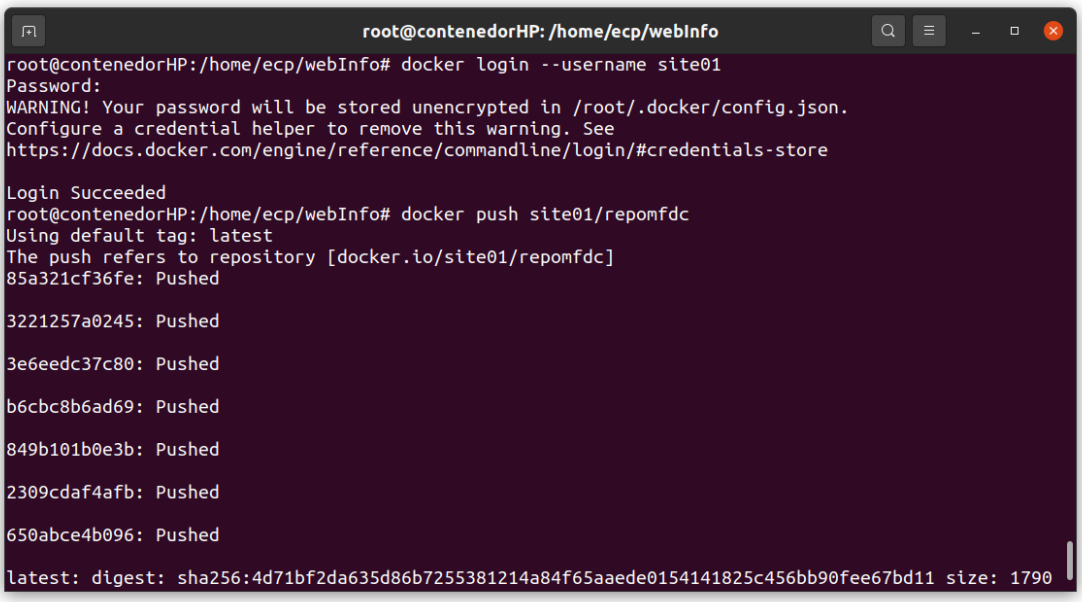
F5: Verificar la capacidad de almacenar imágenes de contenedores en el repositorio.

Acción: se envía al repositorio la imagen de contenedor que se creó para el servicio, utilizando el comando:

```
docker push site01/repomfdc
```

Figura 38

Almacenamiento de imágenes en repositorio



F6: Verificar la capacidad de descargar imágenes del repositorio de contenedores.

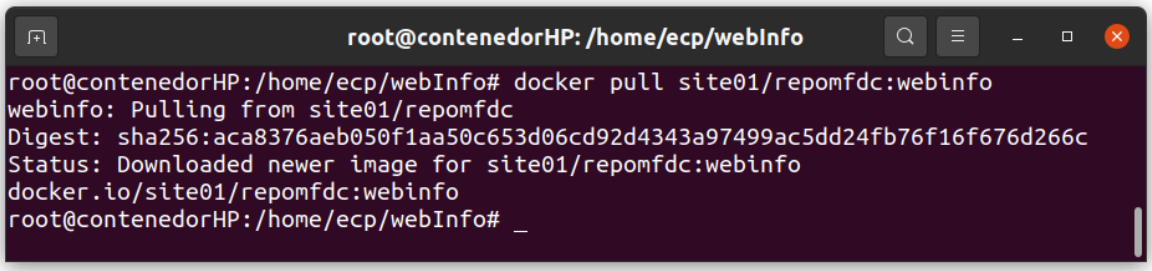
Acción: Se verifica la capacidad de descargar una imagen del repositorio utilizando e comando:

```
docker pull site01/repomfdc:webinfo
```

se aprecia en la imagen el resultado de la operación, que demuestra la capacidad de descargar imágenes desde el repositorio.

Figura 39

Descarga de imágenes de repositorio



Seguidamente, el Especialista en Operaciones solicita las pruebas de seguridad y funcionalidad a los especialistas correspondientes.

PRUEBAS DE SEGURIDAD DE GESTIÓN DE REPOSITORIO

El Especialista en Seguridad conduce las pruebas de seguridad y completa el formato CRS con los resultados.

Conexiones a registros seguras

Se verifica que el sitio web del repositorio de imágenes utiliza SSL, lo que acredita que la conexión es segura y encriptada:

Figura 40

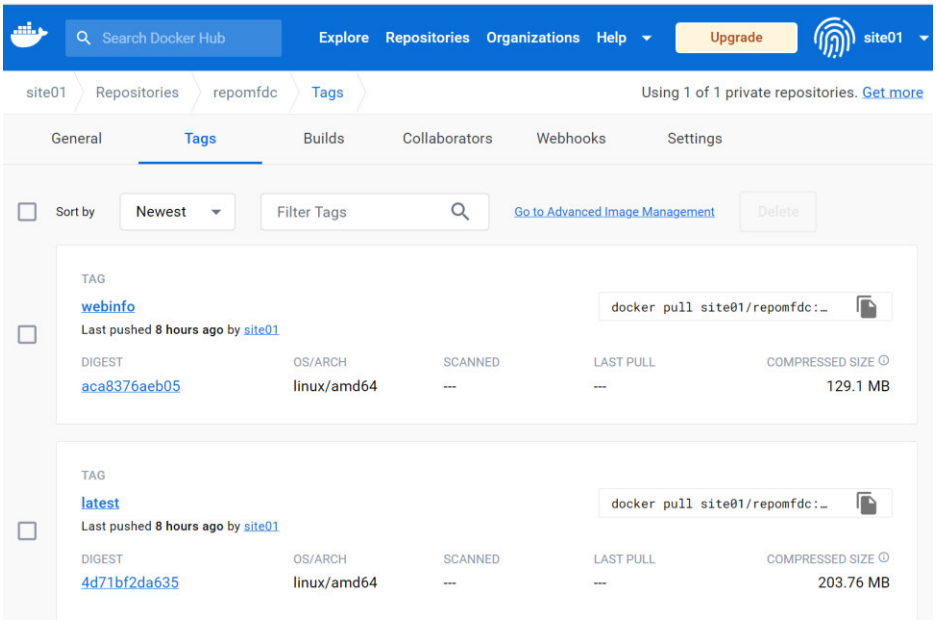
Certificado de proveedor de repositorio

Imágenes de registro vigentes:

Se verifica listando las imágenes almacenadas en el repositorio:

Figura 41

Listado de imágenes en repositorio

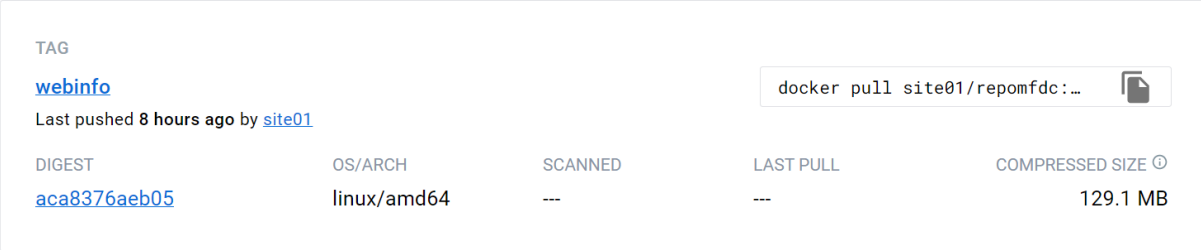


Denominación de imágenes consistente:

Se lista los nombres asignados a las imágenes en el repositorio y se verifica que se ha utilizado la regla de denominación previamente convenida.

Figura 42

Denominación de imágenes de contenedores en repositorio

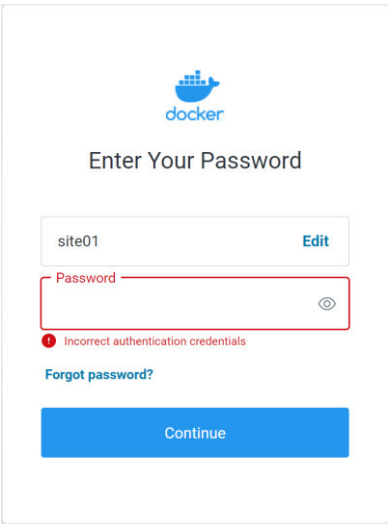


Autenticación y autorización de acceso a imágenes de contenedores:

Se verifica que para acceder al repositorio debe contarse con una cuenta válida.

Figura 43

Control de acceso a repositorio



Los resultados se registran en la ficha CRS:

Figura 44

Resultado de pruebas de seguridad de repositorio

CRS	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES		
PROYECTO	Despliegue de sitio web Portal Docente		
ACTIVIDAD	LISTA DE COMPROBACION DE SEGURIDAD DE REPOSITORIO		
RESPONSABLE	Especialista en seguridad		
FECHA	30/11/2022		
TAREAS		Resultado	Observación
1	Conexiones a registros seguras	SI	
2	Imágenes en registro vigentes	SI	
3	Denominación de imágenes consistente	SI	
4	Autenticación y autorización de acceso a imágenes de contenedores	SI	

PRUEBAS DE FUNCIONALIDAD DE GESTIÓN DE REPOSITORIO

El Especialista en Pruebas completa el formato CRP con los resultados de las pruebas de funcionalidad conducidos.

Disponibilidad de acceso al repositorio:

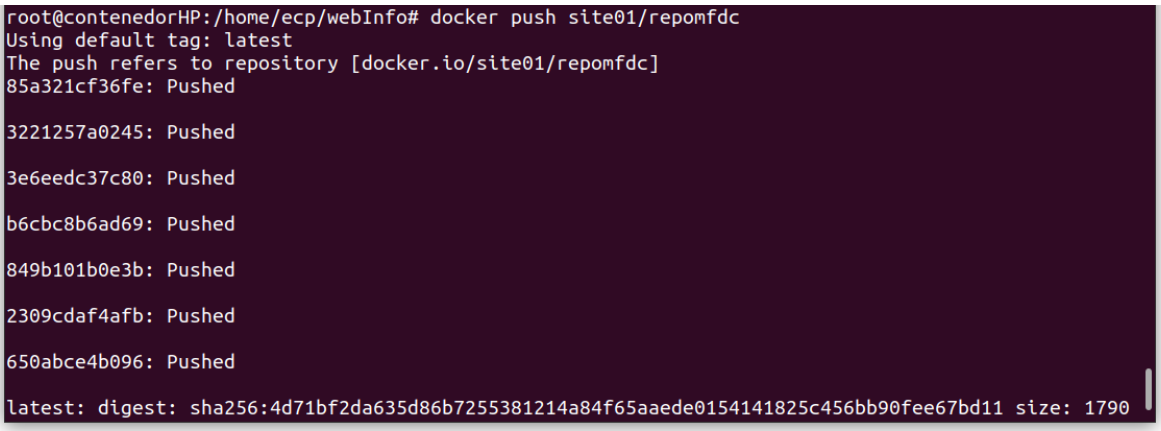
Docker Inc. es una empresa con alcance global, cuyos servicios son utilizados por miler de usuarios a nivel mundial. La infraestructura y servicios que ofrece tienen alta disponibilidad, por lo que la disponibilidad de acceso está garantizada.

Capacidad de almacenamiento de imágenes de contenedores en el repositorio:

Se verifica repitiendo el procedimiento de envío de imágenes al repositorio que se mostró anteriormente.

Figura 45

Envío de imagen de contenedor a repositorio

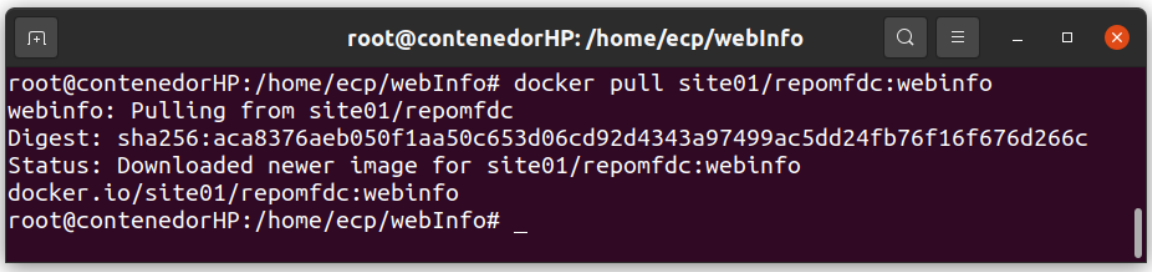


Capacidad de descargar imágenes de contenedores del repositorio:

Se verifica descargando una imagen del repositorio, como se mostró previamente:

Figura 46

Descarga de imágenes de contenedores desde repositorio



Los resultados de las pruebas se registran en la ficha CRP

Figura 47

Resultado de pruebas de funcionalidad de repositorio

CRP	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES		
PROYECTO	Despliegue de sitio web Portal Docente		
ACTIVIDAD	LISTA DE COMPROBACION DE FUNCIONALIDAD DE REPOSITORIO		
RESPONSABLE	Especialista en pruebas		
FECHA	30/11/2022		
TAREAS		Resultado	Observación
1	Disponibilidad de acceso al repositorio.	SI	
2	Capacidad de almacenamiento de imágenes de contenedores en el repositorio.	SI	
3	Capacidad de descargar imágenes de contenedores del repositorio.	SI	

Puesto que las pruebas fueron satisfactorias, se da por concluida la fase de Gestión de Repositorio. El producto final de esta fase es la imagen de contenedor almacenada en el repositorio de imágenes de contenedores seleccionado.

Se procede a la fase final: la fase de Gestión de Despliegue.

C. Gestión de despliegue. La fase de Gestión de Despliegue inicia con la formulación de las opciones de despliegue formuladas por el especialista en operaciones en el formato ED. Las opciones de despliegue permiten especificar características propias del contenedor en ejecución. En el caso del proyecto de prueba, se evalúan las especificaciones del servicio indicadas en el formato ES y se incluyen las pertinentes:

Figura 48

Ficha de especificaciones para el despliegue de sitio web

ED	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO	Despliegue de sitio web Portal Docente	
ACTIVIDAD	ESPECIFICACIONES DE DESPLIEGUE DE SITIO WEB	
RESPONSABLE	Especialista en operaciones	
FECHA	5/12/2022	
ESPECIFICACIONES		
1	Disponibilidad	No crítico
2	Restricciones de modificación	Inmutable
3	Restricciones de acceso	Ninguna

El formato ED es derivado a los especialistas en seguridad y pruebas.

El Especialista en Seguridad formula sus recomendaciones en el formato FDS:

Figura 49

Ficha de recomendaciones de seguridad de despliegue

FDS	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO	Despliegue de sitio web Portal Docente	
ACTIVIDAD	RECOMENDACIONES DE SEGURIDAD DE DESPLIEGUE	
RESPONSABLE	Especialista en seguridad	
FECHA	7/12/2022	
TAREAS		Observación
1	Verificar acceso administrativo al proceso de despliegue	S12
2	Verificar confiabilidad del nodo orquestador	S13
3	Verificar contenedores malintencionados	S14

El Especialista en Pruebas alcanza al especialista en operaciones el formato FDP con sus recomendaciones.

Figura 50

Ficha de recomendaciones de funcionalidad de despliegue

FDP	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO	Despliegue de sitio web Portal Docente	
ACTIVIDAD	RECOMENDACIONES DE FUNCIONALIDAD DE DESPLIEGUE	
RESPONSABLE	Especialista en pruebas	
FECHA	7/12/2022	
TAREAS		Observación
1	Verificar capacidad de acceder al servicio desplegado	F7
2	Verificar la capacidad de controlar el contenedor desplegado	F8
3	Verificar la accesibilidad a bitácoras de monitoreo de servicios	F9

El Especialista en Operaciones prepara el contenedor para el despliegue, tomando en consideración las recomendaciones de seguridad y funcionalidad.

S12: Usar modelo de menor privilegio.

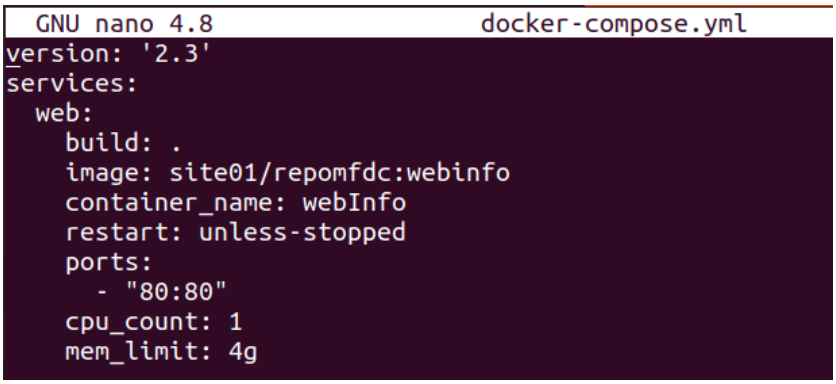
Acción: Se establece como única cuenta para el acceso al repositorio la cuanta que se utilizó para crear el repositorio.

S13: Crear entorno seguro para las aplicaciones ejecutadas.

Acción: Se utilizará Docker-compose para el despliegue del servicio, controlándose el mismo mediante un archivo docker-compose.yml

Figura 51

Archivo de despliegue de contenedor



S14: Instituir entornos de desarrollo, pruebas y producción separados.

Acción: La aplicación de MFDC separa las actividades de desarrollo, pruebas y producción.

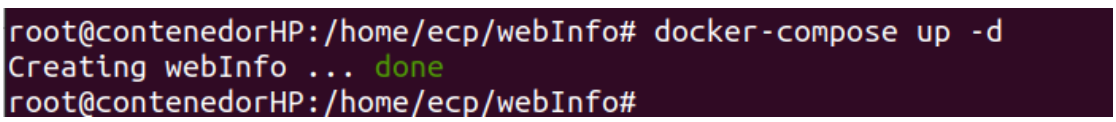
F7: Verificar la capacidad de acceder al servicio desplegado.

Acción: se crea el contenedor a partir de la imagen almacenada en el repositorio mediante el comando

```
docker-compose up -d
```

Figura 52

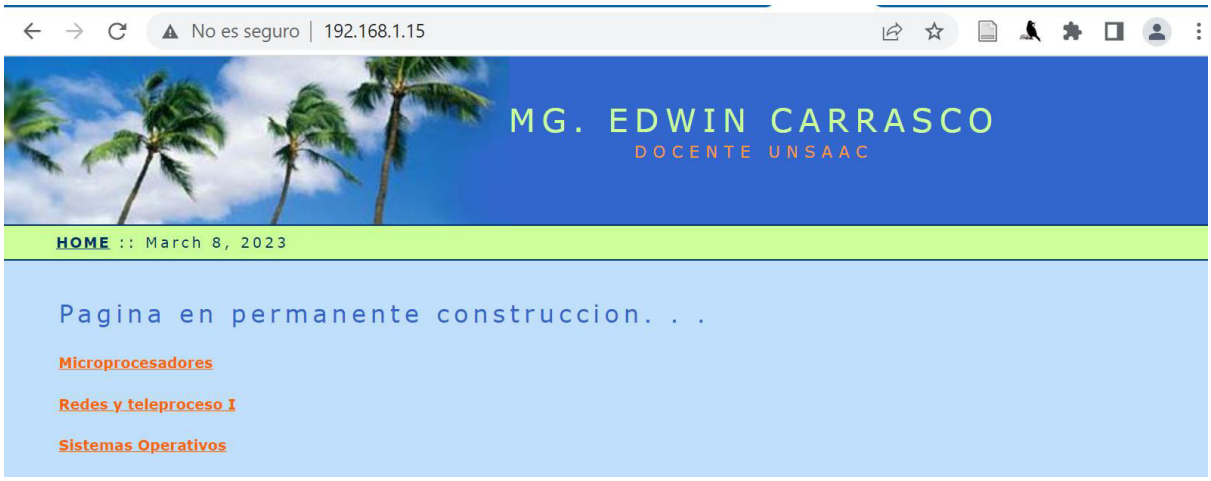
Creación de contenedor



Para verificar la accesibilidad al contenedor desplegado se accede a la dirección del sitio web desde un host externo al sitio de desarrollo. En la siguiente imagen se demuestra el acceso al servidor web desplegado:

Figura 53

Acceso a sitio web desplegado



F8: Verificar la capacidad de controlar el contenedor desplegado.

Acción: Se detiene y reanuda en contenedor con los comandos

```
docker container stop
```

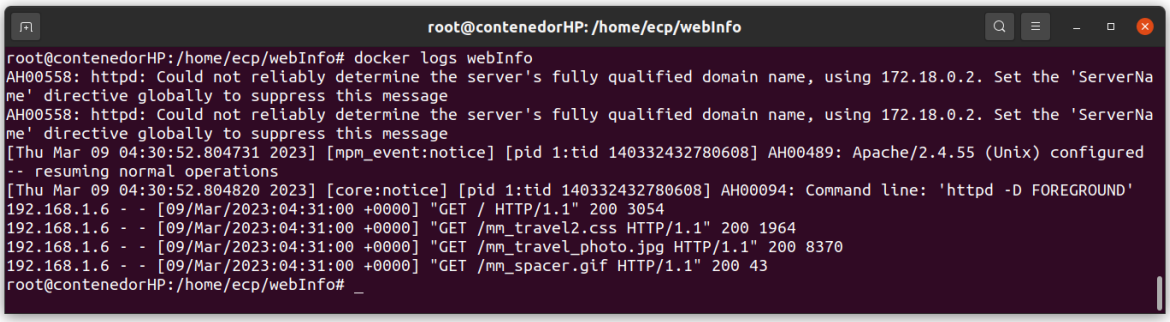
```
docker container start
```

F9: Verificar la accesibilidad a la bitácora de monitoreo de servicios.

Acción: desde la línea de comandos se verifica los registros asociados a la imagen desplegada:

Figura 54

Acceso a logs de contenedor



El Especialista en Operaciones despliega el contenedor en el servidor de producción, donde se realizan las pruebas de seguridad y funcionalidad finales antes de habilitar el servicio para el acceso público.

PRUEBAS DE SEGURIDAD DE GESTIÓN DE DESPLIEGUE

El Especialista en Seguridad realiza las pruebas correspondientes:

Acceso administrativo controlado.

Se verifica que el acceso al repositorio requiere de una cuenta, lo que valida el requerimiento.

Confiabilidad del nodo orquestador.

Como orquestador se utiliza Docker-Compose, el cual es un producto propio de la plataforma Docker. Dadas las características de la plataforma, este requerimiento se considera cumplido.

Contenedores malintencionados.

Los contenedores son instancias de imágenes. La creación de un contenedor a partir de una imagen almacenada en el repositorio de la institución solo puede ser permitido a personal autorizado y registrado en el repositorio institucional con el rol de Colaborador. La asignación de privilegios a los usuarios es potestad del cuerpo administrativo de la RCU. La propia aplicación de MFDC asegura mecanismos para evitar el despliegue de contenedores malintencionados.

Los resultados de las pruebas de seguridad se entregan en el formato CDS.

Figura 55

Resultado de pruebas de seguridad de despliegue

CDS	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES		
PROYECTO	Despliegue de sitio web Portal Docente		
ACTIVIDAD	LISTA DE COMPROBACION DE DESPLIEGUE		
RESPONSABLE	Especialista en seguridad		
FECHA	9/12/2022		
TAREAS		Resultad	Observación
1	Acceso administrativo controlado	SI	
2	Confiabilidad del nodo orquestador	SI	
3	Contenedores malintencionados	NO	

PRUEBAS DE FUNCIONALIDAD DE GESTIÓN DE DESPLIEGUE

El Especialista en Pruebas conduce las pruebas de funcionalidad:

Capacidad de acceso al servicio desplegado:

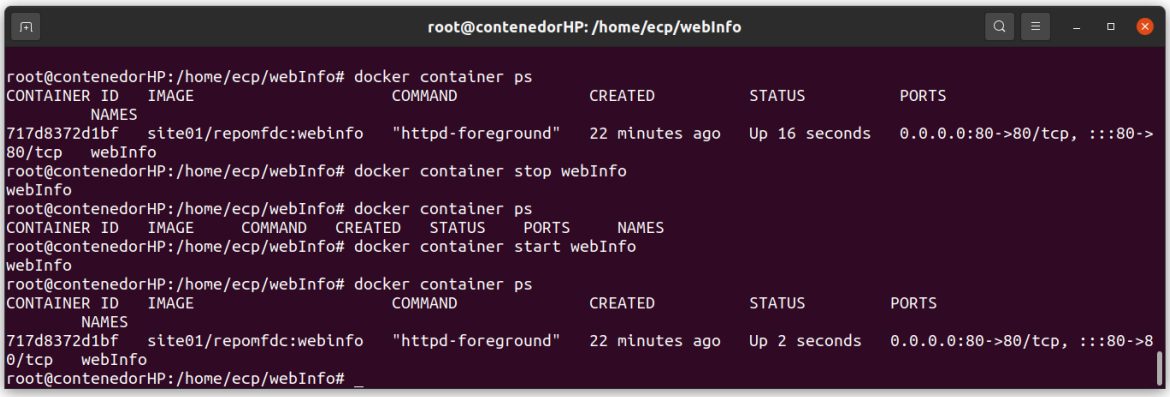
Se verifica mediante el acceso al sitio web desplegado.

Capacidad de controlar el contenedor desplegado:

Se verifica deteniendo e iniciando el contenedor desplegado. Se evidencia mediante la siguiente imagen:

Figura 56

Control de contenedor desplegado

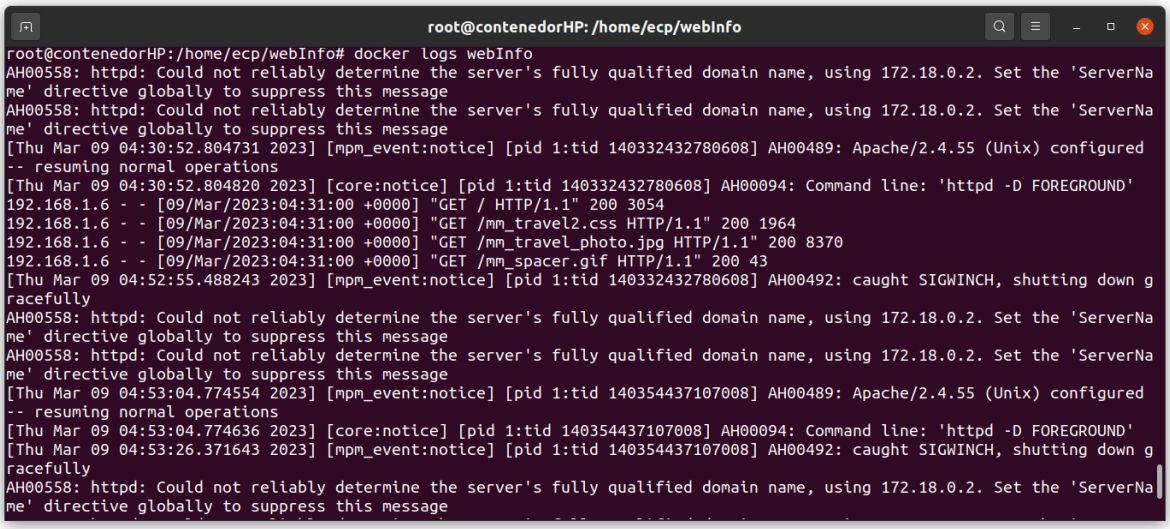


Acceso a bitácoras de monitoreo de servicios.

Se verifica mediante el comando docker log

Figura 57

Verificación de acceso a logs de contenedor



Los resultados de las pruebas de funcionalidad son alcanzados por el Especialista en Pruebas en el formato CDP:

Figura 58

Resultado de pruebas de funcionalidad de despliegue

CDP	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES		
PROYECTO	Despliegue de sitio web Portal Docente		
ACTIVIDAD	LISTA DE COMPROBACION DE FUNCIONALIDAD DE DESPLIEGUE		
RESPONSABLE	Especialista en pruebas		
FECHA	9/12/2022		
TAREAS		Resultado	Observación
1	Capacidad de acceso al servicio desplegado.	SI	
2	Capacidad de controlar el contenedor desplegado.	SI	
3	Acceso a bitácoras de monitoreo de servicios.	SI	

Puesto que las pruebas fueron satisfactorias, el Especialista en Operaciones habilita el acceso al servicio web e informa respecto de la disponibilidad del servicio para el público usuario al administrador web.

Completada la fase de Gestión de Despliegue, culmina la aplicación de la metodología propuesta: MFDC.

Los resultados obtenidos se analizan para la validación del trabajo de investigación en la siguiente sección.

4.3. Resultados de la aplicación de la metodología MFDC

Una vez culminada la aplicación de MFDC, procedemos a realizar pruebas para verificar los indicadores establecidos para el presente trabajo de investigación.

El diseño de pruebas se documenta en el anexo C.

Los resultados de las pruebas se muestran a continuación:

Tabla 12

Resultados de pruebas

	Tiempo de respuesta		Rendimiento del servicio web		Uso de CPU		Uso de memoria	
Nro	Gc	Ge	Gc	Ge	Gc	Ge	Gc	Ge
1	8.383	2.942	181.818	100	3.846	0.078	1.002	0.401
2	3.958	2.808	181.818	100	4.647	0.922	1.002	0.401
3	4.958	3.067	181.818	100	6.12	1.087	1.002	0.401
4	3.75	2.9	76.923	71.429	2.881	0.756	1.002	0.401
5	4.367	2.875	200	181.818	3.392	0.841	1.003	0.401
6	3.558	3.042	200	181.818	3.547	1.167	1.003	0.401
7	4.167	3.033	200	181.818	3.875	0.422	1.003	0.401
8	3.567	2.942	71.429	74.074	3.791	0.836	1.003	0.401
9	3.492	2.925	181.818	200	4.107	0.588	1.003	0.401
10	3.767	2.933	181.818	200	3.778	0.668	1.003	0.401
11	4.133	2.9	181.818	200	3.445	0.921	1.003	0.401
12	4.083	2.883	64.516	74.074	3.697	0.671	1.003	0.401
13	3.817	2.925	153.846	200	3.378	0.838	1.003	0.401
14	3.6	3.025	153.846	200	3.691	1.086	1.003	0.401
15	3.642	2.983	153.846	200	3.713	1.17	1.003	0.401
16	3.358	3.067	74.074	133.333	4.492	0.921	1.003	0.401
17	3.408	2.917	166.667	200	5.921	0.917	1.003	0.401
18	3.525	3.017	166.667	200	3.862	0.671	1.003	0.401
19	3.675	3.117	166.667	200	3.694	1	1.003	0.401

	Tiempo de respuesta		Rendimiento del servicio web		Uso de CPU		Uso de memoria	
Nro	Gc	Ge	Gc	Ge	Gc	Ge	Gc	Ge
20	4.067	2.958	76.923	71.429	3.765	1.001	1.003	0.401
21	3.65	3.125	200	181.818	3.598	0.918	1.003	0.401
22	3.683	3.008	200	181.818	3.535	0.922	1.003	0.401
23	3.767	2.942	200	181.818	3.849	0.836	1.003	0.401
24	3.333	3.058	74.074	68.966	3.043	0.668	1.003	0.401
25	3.392	3.058	200	181.818	3.781	0.84	1.003	0.401
26	3.558	2.95	200	181.818	3.778	0.837	1.003	0.401
27	3.658	2.983	200	181.818	3.706	1.004	1.003	0.401
28	3.617	2.975	76.923	76.923	3.865	1.003	1.003	0.401
29	3.675	3.033	222.222	222.222	3.616	1.003	1.003	0.401
30	3.683	2.925	222.222	222.222	3.056	0.838	1.003	0.401
31	3.875	2.842	222.222	222.222	3.956	0.92	1.003	0.401
32	6.417	2.75	64.516	71.429	3.919	0.918	1.003	0.401
33	3.558	2.858	153.846	166.667	4.26	1.167	1.003	0.401
34	3.425	2.808	153.846	166.667	3.936	0.753	1.003	0.401
35	3.417	2.85	153.846	166.667	6.655	0.753	1.003	0.401
36	3.633	2.883	62.5	76.923	4.264	0.504	1.003	0.401
37	3.258	2.983	142.857	222.222	2.956	0.836	1.003	0.401
38	3.567	3.025	142.857	222.222	3.367	0.671	1.003	0.401
39	3.642	2.992	142.857	222.222	3.296	1.006	1.003	0.401
40	3.783	2.742	64.516	76.923	3.547	0.838	1.003	0.401

	Tiempo de respuesta		Rendimiento del servicio web		Uso de CPU		Uso de memoria	
Nro	Gc	Ge	Gc	Ge	Gc	Ge	Gc	Ge
41	3.5	2.925	142.857	222.222	3.849	1.004	1.003	0.401
42	3.55	2.933	142.857	222.222	3.697	0.587	1.003	0.401
43	3.558	2.95	142.857	222.222	3.456	0.753	1.003	0.401
44	3.6	2.883	66.667	71.429	3.307	0.919	1.003	0.401
45	3.875	3.033	153.846	200	4.103	0.672	1.003	0.401
46	3.908	2.842	153.846	200	3.439	0.668	1.003	0.401
47	3.758	3.075	153.846	200	3.282	1	1.003	0.401
48	3.667	3.233	66.667	74.074	3.384	0.584	1.003	0.401
49	3.8	3.05	153.846	200	3.859	1.172	1.003	0.401
50	3.625	2.875	153.846	200	3.535	0.836	1.003	0.401
51	3.35	3.225	153.846	200	3.939	0.921	1.003	0.401
52	3.517	2.858	66.667	83.333	3.204	0.922	1.003	0.401
53	3.642	2.992	142.857	250	3.451	1.005	1.003	0.401
54	3.408	2.917	142.857	250	3.209	0.503	1.003	0.401
55	3.658	2.767	142.857	250	3.61	0.504	1.003	0.401
56	3.95	2.592	64.516	76.923	3.459	0.67	1.003	0.401
57	3.783	2.383	142.857	200	4.1	0.671	1.003	0.401
58	3.633	2.383	142.857	200	4.34	1.171	1.003	0.401
59	3.842	2.392	142.857	200	4.267	0.668	1.003	0.401
60	3.725	2.742	66.667	71.429	3.208	0.863	1.003	0.401

4.4. Contrastación de hipótesis, análisis e interpretación.

La contrastación de las hipótesis específicas se realiza utilizando los resultados de las pruebas

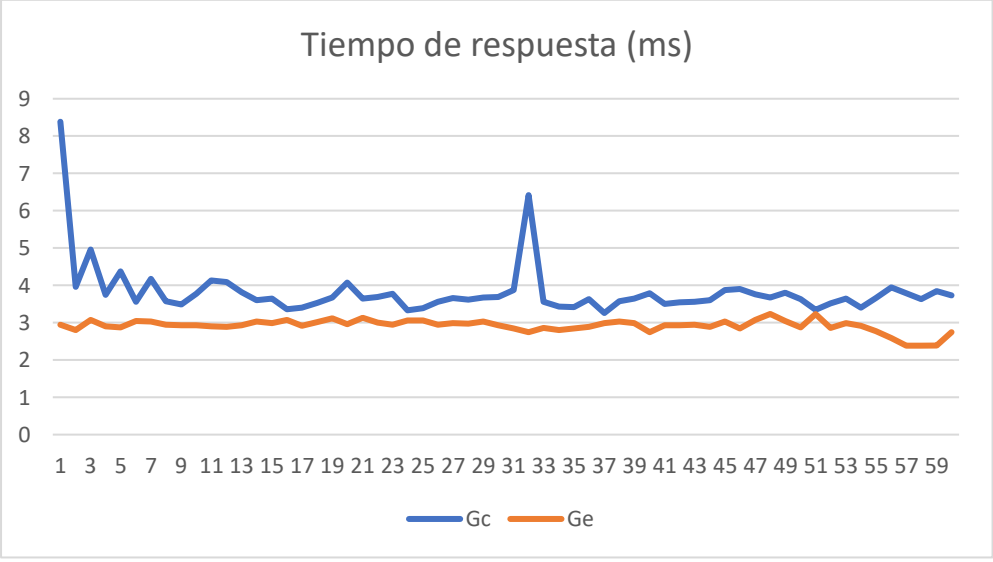
H1: El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC reduce el tiempo de respuesta del servicio web.

ANÁLISIS

La figura 59 muestra los resultados de la evaluación del indicador Tiempo de respuesta del servicio web:

Figura 59

Comparación de resultado de pruebas de Tiempo de respuesta



Los resultados de las pruebas muestran que la aplicación de la metodología para el despliegue del servicio web tiene un efecto positivo en el tiempo de respuesta, es decir, permite reducir el tiempo de respuesta del servicio web evaluado.

Se adjunta pruebas estadísticas para la validación de los resultados de las pruebas:

Tabla 13

Descriptivas de Tiempo de respuesta

	N	Media	Mediana	DE	Mínimo	Máximo
Gc	60	3.82	3.65	0.746	3.26	8.38
Ge	60	2.92	2.94	0.167	2.38	3.23

Tabla 14

Prueba -t para Tiempo de respuesta

			estadístico	gl	p
Gc	Ge	T de Student	8.97	59.0	< .001

Nota. $H_a\ \mu_{\text{Medida 1 - Medida 2}} > 0$

INTERPRETACIÓN

Sea

H : El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC reduce el tiempo de respuesta del servicio web.

H_0 : El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC no reduce el tiempo de respuesta del servicio web.

H_a : El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC reduce el tiempo de respuesta del servicio web.

como $p < 0.001$ para $\alpha = 0.05$, entonces se rechaza H_0 y se acepta H_a.

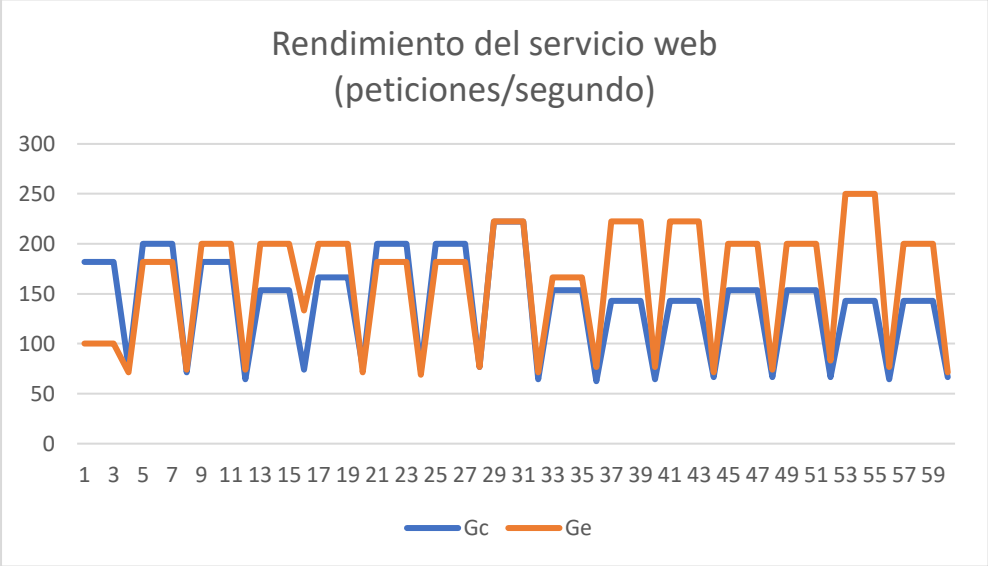
H2: El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC incrementa el rendimiento del servicio web.

ANÁLISIS

Los resultados de las pruebas se muestran y analizan a continuación:

Figura 60

Comparación de resultados de pruebas de Rendimiento del servicio web



Se adjunta pruebas estadísticas para la validación de los resultados de las pruebas:

Tabla 15

Descriptivas de pruebas de rendimiento del servicio web

	N	Media	Mediana	DE	Mínimo	Máximo
Gc	60	144	154	49.1	62.5	222
Ge	60	166	191	58.9	69.0	250

Tabla 16

Prueba - t para análisis de pruebas de rendimiento del servicio web

			estadístico	gl	p
Gc	Ge	T de Student	−3.98	59.0	< .001

Nota. $H_a\ \mu_{\text{Medida 1 - Medida 2}} < 0$

INTERPRETACIÓN

Sea

H : El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC incrementa el rendimiento del servicio web.

H_0 : El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC no incrementa el rendimiento del servicio web.

H_a : El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC incrementa el rendimiento del servicio web.

como $p < 0.001$ para $\alpha = 0.05$, entonces se rechaza H_0 y se acepta H_a.

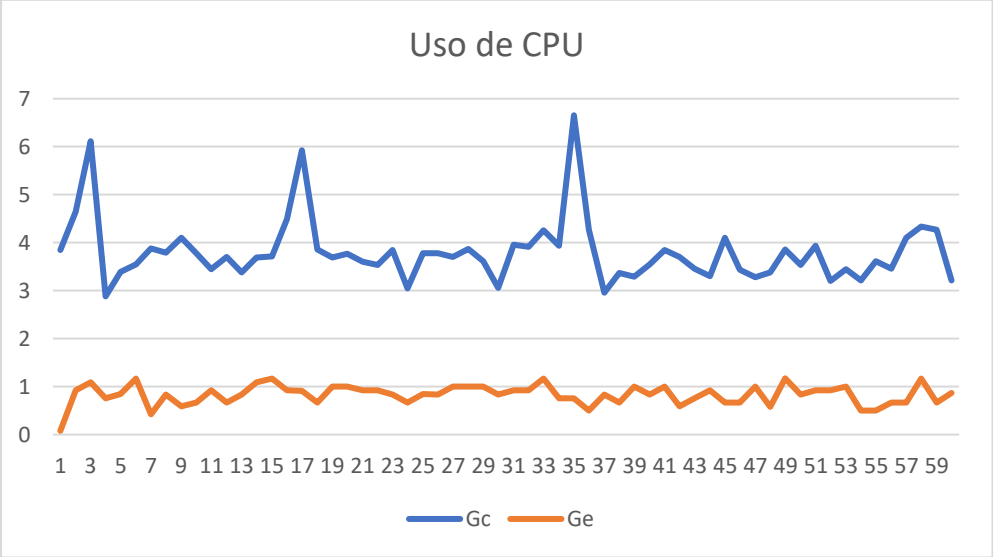
H3: El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC reduce el uso de CPU del servicio web.

ANÁLISIS

Los resultados de las pruebas se muestran y analizan a continuación:

Figura 61

Comparación de resultado de pruebas de Uso de CPU



Las pruebas estadísticas para la validación de los resultados:

Tabla 17

Descriptivas de prueba de Uso de CPU

	N	Media	Mediana	DE	Mínimo	Máximo
Gc	60	3.805	3.702	0.674	2.8810	6.66
Ge	60	0.831	0.839	0.209	0.0780	1.17

Tabla 18

Prueba - t para prueba de Uso de CPU

			estadístico	gl	p
Gc	Ge	T de Student	33.5	59.0	< .001

Nota. $H_a\ \mu_{\text{Medida 1 - Medida 2}} > 0$

INTERPRETACIÓN

Sea

H : El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC reduce el uso de CPU del servicio web.

H_0 : El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC no reduce el uso de CPU del servicio web.

H_a : El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC reduce el uso de CPU del servicio web.

como $p < 0.001$ para $\alpha = 0.05$, entonces se rechaza H_0 y se acepta H_a.

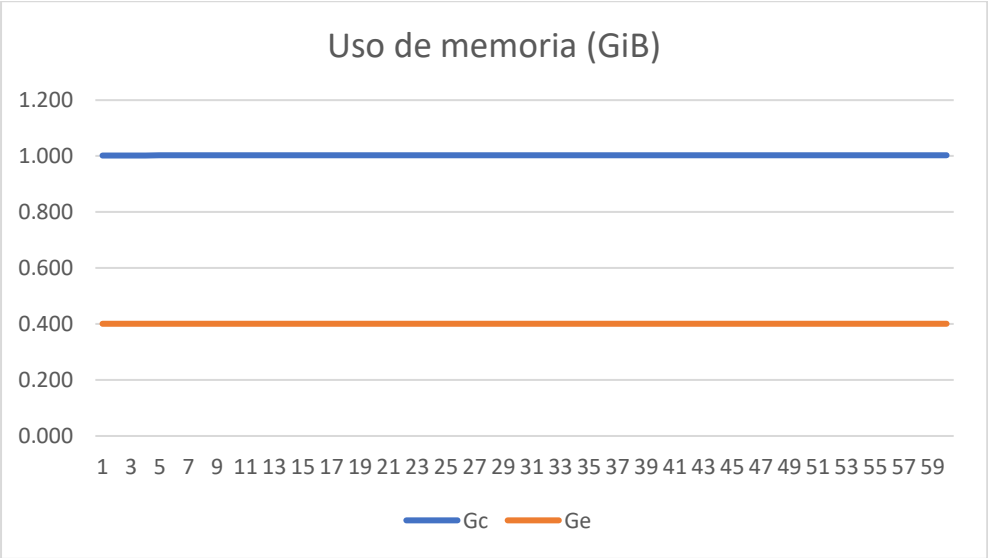
H4: El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC reduce el uso de memoria del servicio web.

ANÁLISIS

Los resultados de las pruebas se muestran a continuación:

Figura 62

Comparación de resultado de pruebas de Uso de memoria



Se adjuntan pruebas estadísticas para la validación de los resultados:

Tabla 19

Descriptivas de prueba de Uso de memoria

	N	Media	Mediana	DE	Mínimo	Máximo
Gc	60	1.003	1.003	2.52e−4	1.002	1.003
Ge	60	0.401	0.401	0.00	0.401	0.401

Tabla 20

Prueba - t para prueba de Uso de memoria

			estadístico	gl	p
Gc	Ge	T de Student	18535	59.0	< .001

Nota. H_a μ Medida 1 - Medida 2 > 0

INTERPRETACIÓN

Sea

H : El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC reduce el uso de memoria del servicio web.

H_0 : El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC no reduce el uso de memoria del servicio web.

H_a : El despliegue de contenedores para proveer el servicio web en la UNSAAC, utilizando la metodología MFDC reduce el uso de memoria del servicio web.

como $p < 0.001$ para $\alpha = 0.05$, entonces se rechaza H_0 y se acepta H_a.

V. DISCUSIÓN DE RESULTADOS

5.1 Discusión

La aplicación de la Metodología Formal de Despliegue de Contenedores – MFDC, ha demostrado que permite mejorar significativamente los indicadores utilizados para medir el nivel de provisión de servicios web.

Seguidamente, se analiza los resultados obtenidos, mismos que se contrastarán con los resultados obtenidos en trabajos de investigación similares.

A. Tiempo de respuesta del servicio web. El tiempo de respuesta del servicio web mejora en un 76.4% respecto de la plataforma actualmente utilizada en la institución donde se aplicó MFDC.

Este resultado es coherente con los resultado de pruebas similares conducidos en el trabajo de investigación sobre costos de virtualización utilizando contenedores, máquinas virtuales y hardware real (Giallorenzo et al., 2021), tanto como en el trabajo de análisis de rendimiento de máquinas virtuales y contenedores Docker expuestos en (Kavitha & Varalakshmi, 2018) y en el trabajo de comparación de rendimiento entre contenedores Linux y máquinas virtuales realizado en (Joy, 2015).

Para explicar el resultado obtenido, debemos de considerar que el servidor web desplegado mediante la tecnología de contenedores se ejecuta utilizando menos capas que las máquinas virtuales, por lo que el flujo de las peticiones y respuestas HTTP tiene una menor sobrecarga.

B. Rendimiento del servicio web. En cuanto al rendimiento del servicio web, las pruebas indican un mayor rendimiento de los servicios web cuando se utiliza MFDC para el despliegue del servidor web. Este resultado es consistente con los resultados obtenidos en el trabajo de evaluación del rendimiento de contenedores Docker y máquinas virtuales presentado en (Potdar et al., 2019), así como en (Joy, 2015).

C. Uso de CPU del servicio web. El uso del CPU en la provisión del servicio web al aplicar MFDC demuestra una mejora respecto de la provisión del servicio web utilizando máquinas virtuales para el despliegue del servidor web en el caso práctico, en un factor de 4.6 en promedio. Desde esta perspectiva, MFDC y la tecnología de contenedores, puede ayudar a utilizar de forma más eficiente los recursos de cómputo de una organización. Por ejemplo, se pueden desplegar más servicios o reducir costos de operación.

Los resultados mostrados, muestran el mismo patrón que los resultados demostrados en (Giallorenzo et al., 2021) y (Potdar et al., 2019).

D. Uso de memoria del servicio web. El uso de memoria es en promedio 2.5 veces mejor en el caso de las pruebas aplicadas al servicio web desplegado utilizando contenedores con MFDC que en el caso del despliegue del servicio web utilizando máquinas virtuales.

Los resultados obtenidos en el presente trabajo coinciden con los que alcanzaron en (Scheepers, 2014) al comparar LXC con Xen y con los descritos en (Bermejo & Juiz, 2019).

Esta diferencia se explica por las características propias de la tecnología de contenedores como es el hecho que los contenedores no necesitan virtualizar hardware ni ejecutar un sistema operativo adicional al sistema operativo nativo.

Como resultado de la aplicación de MFDC y la medición de las variables seleccionadas para su evaluación, se puede verificar que la metodología propuesta es viable y presenta un aporte a la formalización de los procesos de gestión de tecnologías de información y a la promoción y adopción de la tecnología de contenedores para un uso más eficiente de los recursos de TI de las organizaciones.

VI. CONCLUSIONES

a) El proceso de provisión de servicios de red en la UNSAAC, al utilizar la tecnología de contenedores bajo los lineamientos propuestos en la metodología MFDC, mejora.
b) El tiempo de respuesta del servicio web mejora cuando se utiliza la tecnología de contenedores desplegada utilizando la metodología MFDC.
c) El rendimiento del servicio web mejora cuando se utiliza la tecnología de contenedores desplegada utilizando la metodología MFDC.
d) El uso de CPU por parte del servicio web mejora cuando se utiliza la tecnología de contenedores desplegada utilizando la metodología MFDC.
e) El uso de memoria por parte del servicio web mejora cuando se utiliza la tecnología de contenedores desplegada utilizando la metodología MFDC.

VII. RECOMENDACIONES

a) Se recomienda ampliar el alcance de la metodología propuesta para incorporar componentes de gestión y monitoreo de los servicios desplegados en contenedores cuando estos estén en operación.

b) La tecnología de unikernels es una nueva dirección de investigación en alternativas para el despliegue de servicios de red. Se recomienda explorar su viabilidad y la posibilidad de aplicar la metodología propuesta para su uso.

c) Durante el desarrollo del presente trabajo, han surgido metodologías para la gestión de procesos CD/CI considerando aspectos de seguridad. Una de estas metodologías se denomina DevSecOps. Se recomienda contrastar DevSecOps con la presente propuesta.

VIII. REFERENCIAS

Amaral, M., Polo, J., Carrera, D., Mohomed, I., Unuvar, M., & Steinder, M. (2015). Performance Evaluation of Microservices Architectures using Containers. *2015 IEEE 14th International Symposium on Network Computing and Applications*, 27-34.

Balakumar, S., & Kavitha, A. (2018). Docker Container Service for Microservice Applications. *International Journal of Pure and Applied Mathematics, 119*, 15591-15596.

Bermejo, B., & Juiz, C. (2019). Virtual Machine Consolidation: A Systematic Review of its Overhead Influencing Factors. *The Journal of Supercomputing, 76*, 324-361.

Brinkkemper, S., Saeki, M., & Harmsen, F. (1998). Assembly Techniques for Method Engineering. *Proceedings of the 10th International Conference on Advanced Information Systems Engineering (CAiSE'98)*, 381-400.

Bucher, T., Klesse, M., Kurpjuweit, S., & Winter, R. (2007). Situational Method Engineering. On the Differentiation of "Context" and "Project Type". In J. Ralyté, S. Brinkkemper, & B. Henderson-Sellers (Eds.), *Situational Method Engineering: Fundamentals and Experiences* (pp. 33-48). Springer.

Datadog.com. (Junio de 2018). *Datadog.* Datadog: https://www.datadoghq.com/docker-adoption/

Docker Inc. (28 de Enero de 2021). *Docker Inc.* https://docs.docker.com/get-started/overview/

El Amri, A. (2017). *Painless Docker.* Kindle Edition.

Forouzan, B. (2012). *Data Communications and Networking* (Quinta ed.). McGraw Hill.

Giallorenzo, S., Mauro, J., Gyde Poulse, M., & Siroky, F. (2021). Virtualization Costs: Benchmarking Containers and Virtual Machines Against Bare-Metal. *SN Computer Science, 2*, 404-424.

Gonzales-Perez, C. (2007). Supporting Situational Method Engineering with ISO/IEC 24744 and the Work Product Pool Approach. En J. Ralyté, S. Brinkkemper, & B. Henderson-Sellers, *Situational Method Engineering: Fundamentals and Experiences* (Vol. 244, págs. 7-18). Boston: Springer.

Gough, C., Steiner, I., & Saunders, W. (2015). *Energy Efficient Servers.* Apress.

Gutterman, Z., Kolepp, D., Ramirez, E., Sola, J., & Allred, R. (2019). Introduction to Containers, Kubernets, and Red Hat Openshift. (S. Kenlon, D. Sacco, & C. Petlitzer, Eds.)

Huanca, F. (25 de 08 de 2017). Arquitectura para el desarrollo e implementación de servicios web. Puno. http://hdl.handle.net/20.500.12390/1796

Ivanov, K. (2017). Containerization With LXC. Mumbai: Packt Publishing Ltd.

Jayaswal, K. (2006). *Administering Data Centers.* Indianapolis: Wiley.

Joy, A. (2015). Performance Comparison Between Linux Containers and Virtual Machines. *2015 International Conference on Advances in Computer Engineering and Applications*, 342-346.

Kaiser, A. (2018). *Reinventing ITIL in the Age of DevOps.* Apress.

Kane, S., & Matthias, K. (2018). *Docker. Up & Running* (2 ed.). O'Really.

Kavitha, B., & Varalakshmi, P. (2018). Performance Analysis of Virtual Machines and Docker Containers. *Springer Nature*, 99-113.

Khandhar, N., & Shah, S. (2019). Docker - The Future of Virtualization. *2019 IJRAR June 2019, Volume 6, Issue 2*, 164-167.

Krebs, R., Momm, C., & Kounev, S. (2012). Architectural Concerns in Multi-Tenant SaaS Applications. *Proceedings of the 2nd International Conference on Cloud Computing and Services Science*, 426-431.

Krief, M. (2019). *Learning DevOps.* Packt.

Kurose, J., & Ross, K. (2013). *Computer Networking: a top-down approach.* Pearson.

Maguire, K. (2019). Methodology as Personal and Profesional Integrity: Research Designing for Practitioner Doctorates. En C. Costley, & J. Fulton, *Methodologies for Practice Research. Approaches for Professional Doctorates* (págs. 95-115). Sage Publications Ltd.

Miell, I., & Sayers, A. (2016). *Docker in Practice.* New York: Manning.

Negus, C. (2016). *Docker Containers.* Prentice Hall.

Nickoloff, J., & Kuenzli, S. (2019). *Docker in Action* (2 ed.). Shelter Insland: Manning.

Open Container Initiative. (26 de enero de 2021). https://opencontainers.org

Open Container Initiative. (12 de enero de 2021). *Open Container Initiative.* https://opencontainers.org

Pfaller, S., & Hartley, T. (2018). Comparison of Windows and Linux as Docker Hosts. *9th annual conference of Computing and Information Technology Research and Education New Zealand*, (págs. 121-124).

Potdar, A., G, N., Kengond, S., & Mulla, M. (2019). Performance Evaluation of Docker Container and Virtual Machines. *Third International Conference on Computing and Network Communications*, 1421-1428.

Poulton, N. (2020). *Docker Deep Dive. Zero to Docker in a single book.* Publicado independientemente.

Red Hat. (14 de enero de 2021). Introduction to Linux Containers.

Reed, M. (2020). *DevOps. The Ultimate Beginners Guide to Learn DevOps step-by-step.* Publishing Factory.

Rolland, C. (2007). Method Engineering: Trends and Challenges. En J. Ralyté, S. Brinkkemper, & B. Henderson-Sellers, *Situational Method Engineering: Fundamentals and Experiences* (págs. 6-6). Boston: Springer.

Sabharwal, N., & W, B. (2015). *Hands on Docker.* Kindle Edition.

Scheepers, M. J. (2014). Virtualization and Containerization of Application Infrastructure: A Comparison. *21st Twente Student Conference on IT June 23rd*, 7.

Schenker, G. (2020). *Learn Docker - Fundamentals of Docker 19.x* (2 ed.). Birminghan, Mumbai: Packt.

Serrano, B., & Huallpamaita, J. (2018). *Diseño e implementación de una plataforma IoT para la gestión de los controladores semafóricos en la ciudad del Cusco.* Cusco. http://repositorio.unsaac.edu.pe/handle/UNSAAC/4273

Shklar, L. (2009). *Web Application Architecture. Principles, Protocols and Practices* (2 ed.). Wiley.

Souppaya, M., Morello, J., & Scarfone, K. (2017, Setiembre). Application Container Security Guide. Estados Unidos. https://doi.org/https://doi.org/10.6028/NIST.SP.800-190

Stoneman, E. (2020). *Learn Docker in a Month of Lunches.* Manning.

Subraya, B. M. (2006). *Integrated Approach to Web Performance Testing: A Practitioner's Guide.* Covent Garden: IRM Press.

Turnbull, J. (2016). *The docker Book.* Kindle Edition.

UNSAAC. (Febrero de 2009). Manual de Organización y Funciones de la RCU. *Area de Racionalización*. Cusco, Perú.

UNSAAC. (2017). Compendio estadistico N° 32. Cusco, Cusco, Peru.

UNSAAC. (Febrero de 2020). *Universidad Nacional de San Antonio Abad del Cusco*. http://www.unsaac.edu.pe

Zhu, H., & Bayley, I. (2018). If Docker Is The Answer, What Is The Question? . *2018 IEEE Symposium on Service-Oriented System Engineering*, 152-163.

IX. ANEXOS

ANEXO A

CUESTIONARIO PARA RECOPILACIÓN DE DATOS SOBRE PROCESO DE PROVISIÓN DE SERVICIOS WEB EN LA UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO"

PROCESOS:

1. Describa la secuencia de actividades conducentes a la provisión de servicios web en la RCU

2. ¿Cuántas personas participan en los procesos de provisión de servicios web en la RCU?

3. ¿Cuáles son los roles y actividades que cumplen las personas que participan en el proceso de provisión de servicios web?

4. ¿Obedecen, las actividades de provisión de servicios web en la RCU, a alguna metodología?

5. ¿Las actividades de provisión de servicios web en la RCU están establecidas oficialmente mediante algún documento?

6. ¿La RCU aplica alguna norma en los procesos vinculados a la provisión de servicios web?

7. Describa las limitaciones que encuentra en los procesos de provisión de servicios web actualmente utilizado.

TECNOLOGÍA:

1. ¿Qué tecnologías utiliza la RCU para la provisión de servicios web?

2. ¿Cómo ayuda esta tecnología en los procesos de provisión de servicios web?

3. ¿Qué limitaciones encuentra en la tecnología de provisión de servicios web utilizada actualmente?

GESTIÓN:

1. ¿Cómo monitorea el desempeño de los servicios web provistos por la RCU?

2. ¿Cómo está vinculado el sistema de monitoreo con el de gestión?

3. ¿Con que frecuencia realiza ajustes a la configuración de los servidores utilizados por la RCU para proveer el servicio web?

SEGURIDAD:

1. ¿Qué mecanismos de seguridad se usan o consideran el en proceso de provisión de servicios web?

2. ¿Qué políticas de seguridad están establecidas en el proceso de despliegue de servicios web en la RCU?

3. Indique las Normas Técnicas Peruanas que aplica la RCU para la gestión de seguridad de los servicios web

ANEXO B

FORMATOS PARA DOCUMENTACIÓN DE LA METODOLOGÍA MFDC

Formato de recomendaciones de seguridad para la fase de Gestión de imágenes de contenedores:

Figura 63

Formato FIS

FIS	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO		
ACTIVIDAD		
RESPONSABLE		
FECHA		
TAREAS		Observación

Formato para las pruebas de seguridad en la fase de Gestión de imágenes de contenedores

Figura 64

Formato CIS

CIS	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES		
PROYECTO			
ACTIVIDAD			
RESPONSABLE			
FECHA			
TAREAS		Resultado	Observación

Formato para las recomendaciones de seguridad en la fase de Gestión de repositorio

Figura 65

Formato FRS

FRS	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO		
ACTIVIDAD		
RESPONSABLE		
FECHA		
TAREAS		Observación

Formato para las pruebas de seguridad de la fase de Gestión de repositorio

Figura 66

Formato CRS

CRS	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES		
PROYECTO			
ACTIVIDAD			
RESPONSABLE			
FECHA			
TAREAS		Resultado	Observación

Formato para las recomendaciones de seguridad en la fase de Gestión de despliegue

Figura 67

Formato FDS

FDS	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO		
ACTIVIDAD		
RESPONSABLE		
FECHA		
TAREAS		Observación

Formato para las pruebas de seguridad en la fase de Gestión de despliegue:

Figura 68

Formato CDS

CDS	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES		
PROYECTO			
ACTIVIDAD			
RESPONSABLE			
FECHA			
TAREAS		Resultad	Observación

Formato para las recomendaciones para de funcionalidad en la fase de Gestión de imágenes de contenedores:

Figura 69

Formato FIP

FIP	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO		
ACTIVIDAD		
RESPONSABLE		
FECHA		
TAREAS		Observación

Formato para las pruebas de funcionalidad en la fase de Gestión de imágenes de contenedores:

Figura 70

Formato CIP

CIP	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES		
PROYECTO			
ACTIVIDAD			
RESPONSABLE			
FECHA			
TAREAS		Resultado	Observación

Formato para las recomendaciones de funcionalidad para la fase de Gestión de repositorio:

Figura 71

Formato FRP

FRP	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO		
ACTIVIDAD		
RESPONSABLE		
FECHA		
TAREAS		Observaciones

Formato para las pruebas de funcionalidad en la fase de Gestión de repositorio

Figura 72

Formato CRP

CRP	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES		
PROYECTO			
ACTIVIDAD			
RESPONSABLE			
FECHA			
TAREAS		Resultado	Observación

Formato para las recomendaciones de funcionalidad para la fase de Gestión de despliegue

Figura 73

Formato FDP

FDP	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES	
PROYECTO		
ACTIVIDAD		
RESPONSABLE		
FECHA		
TAREAS		Observación

Formato para las pruebas de funcionalidad de la fase de Gestión de despliegue

Figura 74

Formato CDP

CDP	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES		
PROYECTO			
ACTIVIDAD			
RESPONSABLE			
FECHA			
TAREAS		Resultado	Observación

Formato para las especificaciones de sitio web

Figura 75

Formato ES

ES	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES
PROYECTO	
ACTIVIDAD	
RESPONSABLE	
FECHA	
ESPECIFICACIONES	

Formato para las especificaciones de despliegue del sitio web

Figura 76

Formato ED

ED	METODOLOGIA FORMAL PARA EL DESPLIEGUE DE CONTENEDORES
PROYECTO	
ACTIVIDAD	
RESPONSABLE	
FECHA	
ESPECIFICACIONES	

ANEXO C

DISEÑO DE PRUEBAS

Las pruebas realizadas para la validación de la aplicación de MFDC se condujeron utilizando los siguientes equipos y herramientas de software:

Computadores:

Computador para el despliegue del servicio web mediante máquinas virtuales y mediante contenedores:

Tabla 21

Especificaciones de computador

Marca:	Hewlett Packard.
Modelo:	Z Book
CPU:	Intel Xeon E-2176M/2.7GHz
RAM:	15.5GiB
Capacidad de memoria secundaria:	992.2 GB
Interfaz de red: 100Mb/s.	

Terminal de red para pruebas de acceso:

Tabla 22

Especificaciones de terminal de red

Marca:	Genérico.
CPU:	Intel Core i7-4770/3.4GHz

RAM:	16 GB
Capacidad de memoria secundaria:	1 TB
Interfaz de red:	100Mb/s.

Switch de red:

Tabla 23

Especificaciones de switch de red

Marca:	D-Link
Modelo:	DES-1008A.
Número de puertos:	08.
Velocidad:	10/100 Mb/s.

Software:

Tabla 24

Software utilizado en pruebas

Software de virtualización:	VMWare Workstation 17.0
Contenedores:	Docker.
Orquestador:	Docker-Compose.
Herramienta de pruebas del servicio web:	JMeter versión 5.5
Herramientas para escaneo de vulnerabilidades de contenedores:	Secretscanner versión 1.2.0

Herramientas para la validación de archivos Dockerfile:	Dockerfilelint versión 1.8.0
Herramientas para el escaneo de virus, malware y otros	Clamav versión 1.0.1
Software para el procesamiento de datos:	Microsoft Excel versión 2016.
Software para el análisis estadístico de datos:	Jamovi 2.3.21
Sistema operativo para servidores:	Ubuntu 20.04
Sistema operativo para terminal de red:	Windows 10.
Navegador web:	diversos.

Configuración de equipos para las pruebas.

Máquina virtual VMWare:

Tabla 25

Configuración de máquina virtual

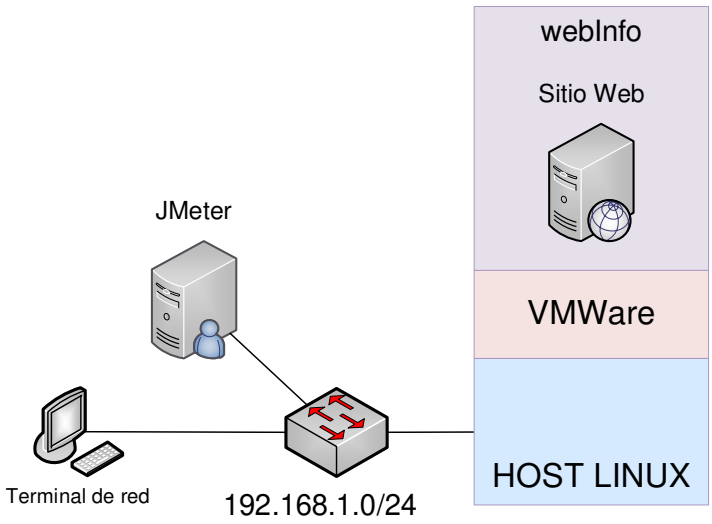
CPU:	01 unidad
RAM:	4GB
Capacidad de memoria secundaria:	250GB.
Interfaz de red:	10/100 Mb/s.

Pruebas.

Configuración del escenario de pruebas para la evaluación del despliegue de servicios web mediante máquinas virtuales.

Figura 77

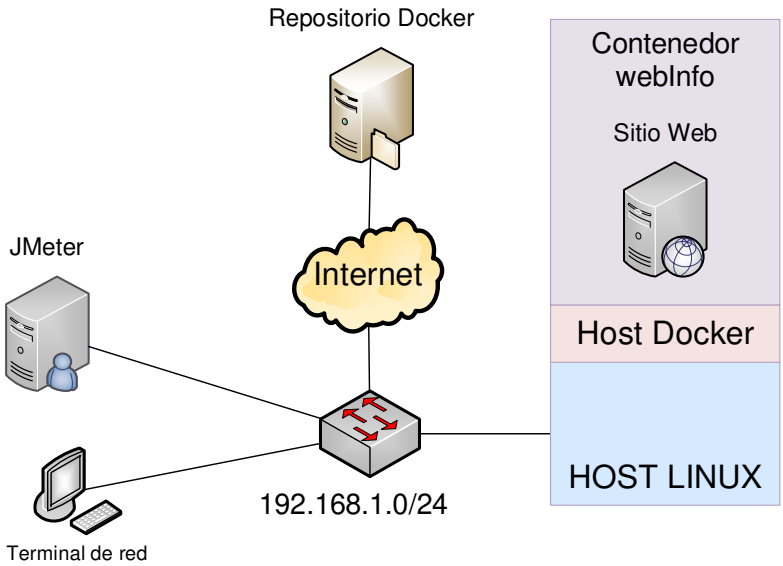
Configuración para pruebas de servicio web con máquinas virtuales



Configuración del escenario de pruebas para la evaluación del despliegue de servicios web mediante contenedores aplicando MFDC.

Figura 78

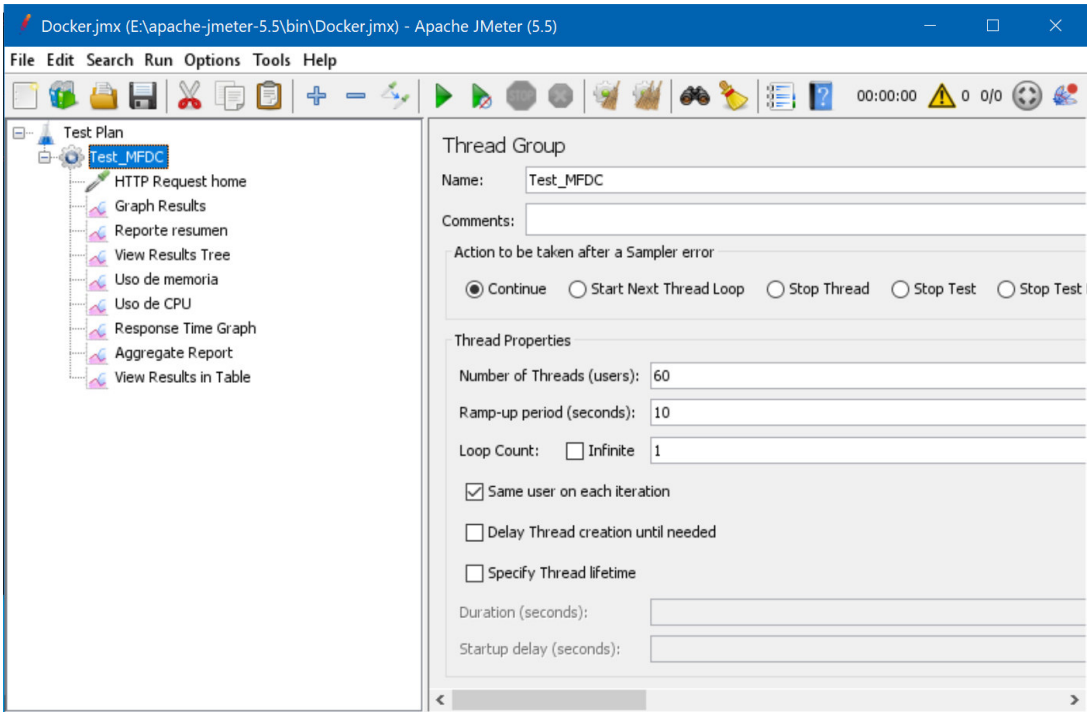
Configuración para pruebas con contenedores y MFDC



Configuración de JMeter.

Figura 79

Configuración de pruebas de carga en JMeter



ANEXO D

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
UNIDAD DE RED DE COMUNICACIONES UNSAAC

"Año de la Unidad, La Paz y El Desarrollo"

CONSTANCIA

La Unidad de Red de Comunicaciones RCU – UNSAAC hace constar, mediante el presente documento, que el Mg. Edwin Carrasco Poblete, desarrollo el trabajo de tesis intitulado "Despliegue De Contenedores Basado En Una Nueva Metodología Para Mejorar El Proceso De Provisión De Servicios Web En La Universidad Nacional De San Antonio Abad Del Cusco" en coordinación con esta dependencia.

Los trabajos realizados comprendieron:

1. Aplicación de cuestionario para recopilación de datos sobre proceso de provisión
2. de servicios web en la Universidad Nacional de San Antonio Abad del Cusco.
3. Elaboración de metodología de despliegue de contenedores de servicios web.
4. Capacitación en tecnología de contenedores al personal de la RCU.
5. Capacitación en aplicación de la metodología propuesta.

La metodología desarrollada se aplicó en un sitio web de prueba lográndose las metas planteadas en el trabajo de tesis.

Por todo lo expuesto, se emite la presente constancia a petición del interesado, para los fines pertinentes.

Cusco, 23 de Junio de 2023

, Atentamente.

Universidad Nacional de San Antonio Abad del Cusco
RED DE COMUNICACIONES - UNSAAC
Johann Mercado León
DIRECTOR(e)

RCU/JML/eac
Cc. Arch.