





Implementación de contenedores dockers en la enseñanza en línea de Ambientes virtualizados

Implementation of Docker containers in online teaching of virtualized environments

Jesús Velázquez Macías¹ , Manuel Veyna Lamas¹ , José Alberto Vela Dávila^{1,2} , Claudia Guadalupe Lara Torres¹ , Juan Antonio González Saenz^{1,2} 

¹ Universidad Politécnica de Zacatecas.
Plan del Pardillo S/N, Parque Industrial, 99050, Fresnillo, Zac.

² Instituto Tecnológico Superior de Fresnillo
Av. Tecnológico #2000, Solidaridad, 99040, Fresnillo, Zac.

*Correo-e: jvelazquez@upz.edu.mx

PALABRAS CLAVE: RESUMEN

Contenedores, docker, testeo, entorno virtualizado, máquina virtual.

La enseñanza de ambientes y escenarios virtualizados en las universidades, es un tema que se complica en la modalidad en línea por su naturaleza práctica, compleja y que requiere de ciertos requisitos de software y hardware para su implementación, los cuales la mayoría de las veces solo están disponibles en los centros de cómputo universitarios, ante la imposibilidad de utilizar estos espacios en tiempos de pandemia surge la necesidad de utilizar otras herramientas tecnológicas que proporcionen la misma experiencia de aprendizaje que una máquina virtual, en el presente trabajo se analiza la implementación de contenedores dockers y de sus principales características como herramienta esencial de integración y estandarización de escenarios y ambientes virtuales independientemente del equipo de cómputo con que cuente el alumno, los cuales son utilizados en tareas de testeo, ejecución y análisis de aplicaciones de software.

KEYWORDS: ABSTRACT

Containers, docker, testing, virtualized environment, virtual machine.

The teaching of virtualized environments and scenarios in universities is a subject that is complicated in the online mode due to its practical, complex nature and that requires certain software and hardware requirements for its implementation, which most of the time only are available in university computer centers, given the impossibility of using these spaces in times of pandemic, the need arises to use other technological tools that provide the same learning experience as a virtual machine, in this work the implementation of containers is analyzed dockers and its main characteristics as an essential tool for the integration and standardization of scenarios and virtual environments regardless of the computer equipment the student has, which are used in testing, execution and analysis of software applications.

Recibido: 16 de julio de 2021 • **Aceptado:** 18 de marzo de 2022 • **Publicado en línea:** 12 de agosto de 2022

1 INTRODUCCIÓN

La enseñanza en línea de materias relacionadas con la Ingeniería en Sistemas computacionales llevan siempre un alto porcentaje de actividades prácticas y presenciales como lo marcan los diferentes planes de estudio, ante la situación actual que se está viviendo en todo el mundo a causa de la pandemia, esta modalidad requiere de nuevas estrategias las cuales garanticen la adquisición de las competencias solicitadas en cada asignatura [1], por desgracia no todos los alumnos cuentan con el equipo de cómputo adecuado para satisfacer estas necesidades[2], por ello se buscan alternativas que sustituyan las actividades que normalmente se realizan en los espacios físicos de las universidades.

Como objeto de estudio en el presente trabajo se muestran el ejemplo aplicado a la materia de Tecnologías de Virtualización, en la Universidad Politécnica de Zacatecas en alumnos de noveno cuatrimestre, en dicha materia y en condiciones normales se utiliza el equipo disponible en los centros de cómputo de la universidad o el equipo con el que cuente el alumno siempre y cuando le permita realizar las actividades solicitadas de acuerdo con sus características y rendimiento.

En esta asignatura se requieren generar ambientes y escenarios virtualizados para realizar tareas de testeo, ejecución y desempeño de aplicaciones de software, así como la instalación de sistemas operativos que trabajen y se comuniquen entre sí, la generación de estos espacios requiere de características especiales como alta capacidad de procesamiento y alta disponibilidad de memoria RAM y espacio en disco suficiente, lamentablemente muchos de los equipos de los alumnos cuentan con cantidades limitadas de las características mencionadas anteriormente [1].

En este punto surge la idea de utilizar los contenedores para reemplazar la virtualización tradicional, ambos conceptos son similares pero no son exactamente lo mismo, los contenedores o dockers en inglés, son más ligeros, en cambio una máquina virtual necesita de un sistema operativo para funcionar [3], mientras que el contenedor utiliza el sistema operativo del equipo tomando solo los recursos básicos los cuales están disponibles de una computadora a otra y de un sistema operativo a otro, es decir solo necesita de características esenciales para funcionar, estas características se empaquetan en el contenedor haciéndolas disponibles independientemente de la maquina en la que se utilicen.

El diseño lógico de un contenedor sigue los siguientes pasos

- Diseño de la aplicación.
- Desarrollo de la aplicación (codificación).

- Implementar el código en un entorno de prueba.
- Empaquetar la aplicación.

Lo que hace ligero a un contenedor es precisamente su capacidad de solo tomar aspectos esenciales de funcionamiento del sistema operativo por ello la ventaja sobre una máquina virtual tradicional [4]. Al estar empaquetados todos los recursos que necesita para ejecutarse puede ser portado a otras computadoras y sistemas operativos diferentes a donde fue creado originalmente [5], siempre y cuando tengan instalada la aplicación Docker, que es la que gestiona, mantiene y da soporte a los contenedores [6].

La aplicación Docker está disponible desde su página oficial y cuenta con soporte para distintos sistemas operativos además de una amplia documentación que facilita su implementación [7].

2 METODOLOGÍA

Alineados al manual de asignatura de la materia Tecnologías de Virtualización diseñado por la Coordinación de Universidades Tecnológicas y Politécnicas para la carrera de Ingeniería en Sistemas Computacionales, se empataron los siguientes temas haciendo uso de contenedores como sustitución a los temas originales, también se especifica el apartado correspondiente en la documentación oficial de Docker[8].

La siguiente tabla muestra las equivalencias implementadas para la Unidad 1

Tabla 1. Equivalencias implementadas para la unidad 1

TEMAS	Actividad del plan original	Actividad sustituida en relación a contenedores
Marco conceptual 1	<p>Definir el concepto de virtualización.</p> <p>Describir las características de la virtualización.</p> <p>Definir los conceptos:</p> <ul style="list-style-type: none"> - Huésped. - Anfitrión. - Emulación / Simulación. - Máquina virtual. - Técnica de virtualización. - Hypervisor. <p>Describir el funcionamiento de las Máquinas Virtuales.</p> <p>Identificar firmas comerciales y no comerciales proveedoras de recursos de virtualización. Explicar el proceso de selección de recursos de virtualización.</p>	<p>Definir el concepto de contenedor.</p> <p>Describir las características de las principales funciones de los contenedores.</p> <p>Definir los conceptos:</p> <ul style="list-style-type: none"> - Host. - Huesped. - Encapsulación. - Kernel. - Modularidad. <p>Describir el funcionamiento de un contenedor.</p> <p>https://docs.docker.com/get-started/</p> <p>Identificar los sistemas operativos que son compatibles con la tecnología de Docker</p> <p>https://docs.docker.com/get-docker/</p>

La siguiente tabla muestra las equivalencias implementadas para la Unidad 2

Tabla 2. Equivalencias implementadas para la unidad 2

TEMAS	Actividad del plan original	Actividad sustituida en relación a contenedores
Virtualización de recursos	<p>Definir el concepto de virtualización de recurso.</p> <p>Describir el proceso de virtualización de los recursos:</p> <ul style="list-style-type: none"> - Kernel. - SO invitado. - CPU. - Memoria. - Almacenamiento. - Entrada / Salida. - Red. - Terminal de sistema. 	<p>Definir el concepto de implementación basado en imágenes.</p> <p>Describir las características del repositorio de docker y como este almacena imágenes de contenedores para poder compartirlos entre varios equipos.</p> <p>https://docs.docker.com/docker-hub/</p>
Ambientes de prueba	<p>Describir los procesos de configuración de las máquinas virtuales:</p> <ul style="list-style-type: none"> - Plantilla de Máquina Virtual. - Clon de Máquina Virtual. - Reutilización de configuraciones. - Personalización de despliegue de Máquina Virtual. <p>Describir el proceso de configuración de parámetros de los entorno operativo</p>	<p>Describir el proceso de instalación de Docker en los distintos sistemas operativos.</p> <p>https://docs.docker.com/get-started/</p> <p>Realizar un contenedor de prueba para comprobar sus características y funcionalidad, utilizando la plantilla "Docker for beginners"</p> <p>https://docs.docker.com/samples/</p> <p>https://github.com/docker/labs/tree/master/beginner/</p>

La siguiente tabla muestra las equivalencias implementadas para la Unidad 3

Tabla 3. Equivalencias implementadas para la unidad 3

TEMAS	Actividad del plan original	Actividad sustituida en relación a contenedores
Control de acceso	<p>Definir los conceptos:</p> <ul style="list-style-type: none"> - Control de acceso. - Perfil de usuario. - Grupos de usuarios. - Esquema de privilegios. - Cortafuegos. - Parámetros de seguridad de servicios. - Parámetros de alerta y códigos de error. <p>Describir el proceso de configuración de los parámetros de seguridad de entornos operativos heterogéneos.</p>	<p>Describir el proceso de implementación de seguridad en contenedores basado en "namespaces"</p> <p>https://docs.docker.com/engine/security/</p> <p>Describir el proceso de implementación de seguridad en contenedores basado en "Control groups"</p> <p>https://docs.docker.com/engine/security/</p>
Alta disponibilidad y seguridad de los datos	<p>Definir los conceptos:</p> <ul style="list-style-type: none"> - Alta prestación. - Alta disponibilidad. - Indicador óptimo de rendimiento según recurso. - Tolerancia a fallos. - Protocolos de replicado. - Protocolos de migración. - Protocolos de restauración. <p>Describir el proceso de optimización de la alta disponibilidad y seguridad de datos en entornos computacionales heterogéneos.</p>	<p>Describir el proceso que realiza un contenedor para garantizar la seguridad y disponibilidad de la información por medio del "Docker Engine API"</p> <p>https://docs.docker.com/engine/api/</p>
Mantenimiento y supervisión	<p>Describir los elementos de mantenimiento y supervisión de máquinas virtuales:</p> <ul style="list-style-type: none"> - Herramientas de diagnóstico fuera de línea. - Herramientas de generación de informes. - Actualización de servicios manual remoto u automático. - Scripts de depuración y 	<p>Describe las buenas practicas de programación utilizando Docker y menciona como ayudan al diseño de aplicaciones confiables, livianas y modulares.</p> <p>https://docs.docker.com/develop/dev-best-practices/</p>

La siguiente tabla muestra las equivalencias implementadas para la Unidad 4

Tabla 4. Equivalencias implementadas para la unidad 4

TEMAS	Actividad del plan original	Actividad sustituida en relación a contenedores
Introducción al cómputo paralelo	<p>Definir los conceptos:</p> <ul style="list-style-type: none"> - Arquitectura paralela. - Arquitectura distribuida. - Procesos. - Región paralela. - Supercomputadoras y Clusters. <p>Describir los procesos de la computación paralela:</p> <ul style="list-style-type: none"> - Descomposición de tareas. - Organización de memoria. <ul style="list-style-type: none"> - Compartida. - Distribuida. - Comunicaciones: <ul style="list-style-type: none"> - Punto a punto - Colectiva - Bloqueantes y no bloqueantes <p>Describir las características de los entornos de cómputo paralelo.</p> <p>Explicar el método de diseño de los entornos de cómputo paralelo.</p>	<p>Describir los procesos a seguir para ejecutar una aplicación en la nube utilizando el servidor Azure.</p> <p>https://docs.docker.com/cloud/aci-integration/</p>
Interfaces Open MP, MPI	<p>Describir los elementos de las interfaces de programación OpenMP y MPI.</p> <p>Describir los procesos de programación en OpenMP y de MPI</p> <ul style="list-style-type: none"> - Mecanismos de sincronización. - Tasking. 	<p>Describir el funcionamiento de BuildKit para la creación de imágenes de contenedores en distintas arquitecturas y con distintos parámetros</p> <p>https://docs.docker.com/develop/develop-images/build_enhancements/</p>

3 RESULTADOS

En la implementación de las modificaciones expresadas en este documento sobre el manual original se obtuvo un nivel de aprobación por parte de los estudiantes de un 93.33% en la asignatura de Tecnologías de Virtualización el cual está asentado en el acta de calificaciones correspondiente, comparado con el mismo periodo del año 2019 en el cual se puso en práctica el manual de la materia al pie de la letra sin modificaciones, el porcentaje de aprobación presenta resultados significativamente inferiores a los aplicados en el presente trabajo (73.91%).

Con ello se establece que la adquisición de las competencias establecidas en el manual de asignatura fue cumplida incluso por encima de los resultados obtenidos en años y cuatrimestres anteriores, adicionalmente se detectó que el 100% de los equipos de cómputo de los alumnos corrieron satisfactoriamente el programa Docker independiente de sus características de software y hardware, con lo que se comprueba uno de las principales bondades de Docker la cual es la ligereza y portabilidad.

4 CONCLUSIONES

La labor docente es innovar en la impartición de educación con los medios que estén disponibles y más aun estos tiempos tan cambiantes donde la educación se vuelve dinámica.

Docker ha cambiado la forma de trabajo de los desarrolladores en todo el mundo [9], esta es una oportunidad para aprovechar recurso gratuitos y de fácil acceso para el público en general pero principalmente para los alumnos obteniendo herramientas para su futuro laboral y profesional.

La premisa de Docker menciona que sus contenedores se "pueden ejecutar en cualquier máquina, en cualquier infraestructura y en cualquier nube". La portabilidad, flexibilidad y simplicidad que esto permite son las razones fundamentales que explican por qué Docker ha sido capaz de crear tanta captación en tan poco tiempo. [10]

Los puntos más fuertes que destacan el uso de contenedores diseñados con Docker son la portabilidad, el aislamiento, la seguridad y el ahorro de tiempo, por lo que facilita el traslado entre sistemas operativos, desarrollo y testeo de aplicaciones independientemente de el software de programación.

REFERENCIAS

- [1] M. J. Vidal Ledo, M. de la C. Barciela González Longoria, y I. Armenteros Vera, “Impacto de la COVID-19 en la Educación Superior”, *Educ. Médica Super.*, vol. 35, núm. 1, 2021.
- [2] J. L. Abreu, “Tiempos de Coronavirus: La Educación en Línea como Respuesta a la Crisis.”, *Rev. Daena Int. J. Good Conscience*, vol. 15, núm. 1, 2020.
- [3] J. A. Ibarra Báez y J. E. Rojas Coppari, “Comparación de rendimiento entre contenedores Docker y máquinas virtuales”, PhD Thesis, Facultad Politécnica, Universidad Nacional del Este, 2015.
- [4] E. Casalicchio y V. Perciballi, “Measuring docker performance: What a mess!!!”, en *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion*, 2017, pp. 11–16.
- [5] “Contenedores de Docker | ¿Qué es Docker? | AWS”, *Amazon Web Services, Inc.* <https://aws.amazon.com/es/docker/> (consultado jul. 16, 2021).
- [6] B. B. Rad, H. J. Bhatti, y M. Ahmadi, “An introduction to docker and analysis of its performance”, *Int. J. Comput. Sci. Netw. Secur. IJCSNS*, vol. 17, núm. 3, p. 228, 2017.
- [7] F. A. Lopez Choque, “IMPLEMENTACIÓN DE CONTENEDORES DOCKER EN APLICACIONES DE SOFTWARE”, PhD Thesis, 2019.
- [8] L. Torrico Rojas, “COMPARAR ‘MAQUINAS VIRTUALES VS DOCKER’”, PhD Thesis, 2019.
- [9] J. Turnbull, *The Docker Book: Containerization is the new virtualization*. James Turnbull, 2014.
- [10] “Docker Desktop for Windows user manual”, *Docker Documentation*, jul. 13, 2021. <https://docs.docker.com/docker-for-windows/> (consultado jul. 15, 2021).

Acerca de los autores



MANUEL VEYNA LAMAS

El MIA Manuel Veyna Lamas es Ingeniero en Comunicaciones y Electrónica por la Universidad Autónoma de Zacatecas, y cursó la Maestría en Informática Administrativa en la Universidad Autónoma

de Durango. Cuenta también con una Especialidad en Computación y un diplomado como Instructor de Inglés. Trabajó 9 años y medio en la industria privada como programador y analista de sistemas en una empresa con presencia mundial. Posteriormente se incorporó a la docencia en la Universidad Politécnica de Zacatecas donde labora desde 2006.

Su área de trabajo es la programación, enfocándose más en Sistemas de Información, Dispositivos Móviles y Hardware. Ha publicado varios artículos en congresos acerca de estos temas. Actualmente tiene otros proyectos en estas las líneas de investigación de su Cuerpo Académico.



JOSÉ ALBERTO VELA DÁVILA

Obtuvo el grado de Maestro en Ingeniería de Software por parte del Centro de Investigación en Matemáticas (CIMAT) campus Zacatecas en el año 2008. Su trabajo de investigación fue

“Ambientes de Desarrollo de Software basado en componentes”. Actualmente es aspirante al Doctorado en Ciencias Computacionales del CIMAT, en la línea de investigación de Human Computer Interaction (HCI).

Se desempeña como docente del programa Educativo de Ingeniería en Sistemas Computacionales en el Instituto Tecnológico Superior de Fresnillo y la Universidad Politécnica de Zacatecas ambas instituciones ubicadas en la ciudad de Fresnillo, Zacatecas, México.



JUAN ANTONIO GONZALEZ SAENZ

Formación: Licenciatura en Ingeniería en Sistemas Computacionales y Posgrado Maestría en Ciencias de la Computación por el ITESM. Profesor de Tiempo Completo Investigador de la

Universidad Politécnica de Zacatecas, perteneciente a la academia de Ingeniería en Sistemas Computacionales, con 15 años de experiencia en materias relacionadas con las Ciencias Computacionales.

Profesor de Asignatura del Tecnológico Nacional de México sede Fresnillo, Zac., impartiendo materias relacionadas con las ciencias de la Probabilidad y la Estadística, y algunas otras ramas de estudios como la Investigación de Operaciones y las Matemáticas Aplicadas.

Asesor técnico en la industria privada en el campo de Gestión de Procesos y Control Estadístico de la Calidad.



Claudia Guadalupe Lara Torres

Nació en la ciudad de Zacatecas el 30 de Octubre de 1980, es Ingeniero Industrial y tiene la Maestría en Administración, ambas realizadas en el Instituto Tecnológico campus Zacatecas, actualmente está elaborando una tesis para obtener el grado de Doctor en Administración

por la Facultad de Contaduría de la Universidad Autónoma de Zacatecas.

Dentro del ámbito laboral ha trabajado en el sector productivo durante 7 años, y aproximadamente 12 años en la educación. Actualmente trabaja en la Universidad Politécnica de Zacatecas como profesor de tiempo completo, realizando funciones como la docencia, asesoría, gestión, tutoría e investigación.

Ha participado en congresos nacionales con la publicación de artículos y ponencias en ciudades como Aguascalientes, Celaya, Morelia, Zacatecas y Durango, e internacionales en países como Colombia y Cuba.