

# VIRTUALIZACIÓN Y CONTENEDORES

MARKO ANTONIO RIVAS RIOS

JORGE LUIS MAMANI MAQUERA

ORLANDO ANTONIO ACOSTA ORTIZ

YOFER NAIN CATARI CABRERA

ORESTES RAMIREZ TICONA

ROBERTO ZEGARRA REYES

Universidad Privada de Tacna

Mayo 13, 2019

## Abstract

*En este trabajo se realizará un estudio de las tecnologías de virtualización de contenedores con el fin de implementar y poner en marcha un sistema que permita orquestar el despliegue de aplicaciones sobre un entorno empresarial. Para ello, se realizará en primera instancia un análisis de los sistemas de virtualización más habituales para continuar introduciendo los conceptos y sistemas de virtualización de contenedores. Una vez introducida la parte teórica se analizan distintas herramientas de virtualización de contenedores centrándonos en la herramienta Docker para la cual se detalla su arquitectura, funcionamiento y proceso de instalación para finalizar con un par de ejemplos prácticos de despliegue de servicios. A continuación, una vez que ya hemos implementado y analizado un sistema de virtualización de contenedores como tecnología necesaria de base, pasamos a estudiar distintas soluciones del mercado para implementar un sistemas de orquestación basado en microservicios para el despliegue de aplicaciones de carácter corporativo. Finalizamos con la implantación, instalación y puesta en marcha del sistema estudiado acompañado de unos ejemplos de orquestación usando dos aplicaciones de código abierto que se ven bastante habitualmente en los entornos corporativos actuales para dar soporte a distintas soluciones.*

## I. INTRODUCCIÓN

LA tecnología llegó para complementar y completar la virtualización de servidores es la virtualización de contenedores de aplicaciones. Esta tecnología va un paso más allá en el paradigma de la virtualización, permitiendo no sólo el salto de virtualizar servidores sino también de virtualizar directamente un contenedor donde se ejecuta una aplicación, permitiendo de este modo una mayor abstracción aislando la componente "lógica de la aplicación" del componente "sistema operativo".

## II. OBJETIVOS

SE busca saber un poco más sobre:  
**¿Qué es la virtualización?**

- Técnica que permite, mediante software, convertir una máquina real en varias máquinas independientes que se ejecutan al mismo tiempo.
- Cada una de estas máquinas independientes recibe el nombre de "máquina virtual huésped", y puede estar gestionada

por cualquier sistema operativo (o casi).

- La máquina real sobre la que se ejecutan los huéspedes se denomina “máquina anfitriona”. - El software que hace posible la ejecución simultánea de varios huéspedes sobre un único anfitrión es el “hipervisor”.

### ¿Qué es la virtualización en contenedores?

---

- Cada huésped “verá” su propia CPU, memoria, discos, etc, independientemente de los recursos de que disponga el anfitrión o el resto de huéspedes.

### ¿Qué se consigue con la virtualización?

---

- En centros de datos:

- Integrar en una única máquina varios servidores; se ahorra en: adquisición de equipos, actualización de hardware, costes de mantenimiento (refrigeración, por ejemplo), espacio, etc.
- Se mantiene entre máquinas el mismo nivel de aislamiento que si éstas estuviesen físicamente separadas (un fallo en una máquina virtual no afecta al resto).

### ¿Qué se consigue con la virtualización en contenedores?

---

- En entornos personales:

- Lograr, con una única máquina, separar varios entornos de desarrollo y/o pruebas.

- Aprender a utilizar nuevo software o nuevas técnicas sin poner en riesgo el software preexistente en nuestro equipo. Ejemplos:

- Instalar sobre Windows una máquina virtual VMWare que sea un Linux Centos con Oracle express instalado.
- Montar y probar una red local formada por varias máquinas virtuales sobre un

único anfitrión.

## Virtualización

---

La virtualización es una técnica que permite abstraer, por medio de herramientas SW, el HW proporcionado por una máquina física (anfitriona) para simular el funcionamiento de otra máquina (huésped). Al conjunto de la máquina huésped y su SO propio se le llama máquina virtual, la cual utiliza los recursos de la máquina física anfitriona sin ser la realmente poseedora de los mismos. Esto implica ciertas ventajas potenciales como economía de recursos, ahorro de espacio físico, coexistencia de múltiples SO en una única máquina, mayor aislamiento entre procesos, etc.

## Contenedor

---

Docker es un proyecto que permite crear aplicaciones en contenedores de software que sean ligeras, portátiles y autosuficientes. Los contenedores son paquetes de elementos que te permiten crear un entorno donde correr aplicaciones independientemente del sistema operativo del host. El propósito de este proyecto es investigar uso de la virtualización de servidores basada en Docker Containers, entender la tecnología que se ejecuta detrás de ella, conocer las posibilidades que tiene y construir un clúster de contenedores que ejecuten una aplicación.

## Tipos de virtualización

---

La virtualización permite la consolidación de múltiples recursos de TI, eleva los índices de utilización de servidores y almacenamientos, ahorra costos y espacios y permite elevar los índices de disponibilidad.

**Los Sigüientes son tipos de virtualización:**

- Virtualización de Servidor
- Virtualización de Almacenamiento
- Virtualización de Escritorio
- Virtualización de Redes
- Virtualización de Redes

### **Ventajas de la Virtualización**

---

- Disminuye la utilización de hardware físico.
- Reducción de gastos. Al disminuir los hardwares físicos los gastos asociados a ellos (luz, mantenimiento, etc.) se ven recordados.
- Aumento de la eficiencia. A medida que la virtualización se va estableciendo dentro de una compañía, los usuarios utilizarán más eficiente los componentes del hardware y, por tanto, no será necesario establecer diferentes conexiones de internet para servidores, ordenadores e emails.
- Largo ciclo de vida. Con la virtualización los programas se almacenan en servidores, lo que implica que la necesidad de equipos más modernos es más reducida que en un hardware.

## **III. VIRTUALIZACIÓN Y CONTENEDORES**

### **i. Virtualización**

#### **A). Modelos principales de virtualización**

##### **1. Virtualización de plataforma**

---

Este es un modelo especialmente a tener en cuenta, ya que es el aplicado para lo que se llama consolidación de servidores. La virtualización o consolidación de servidores puede verse como un particionado de un servidor físico de manera que pueda albergar distintos servidores dedicados (o privados) virtuales que ejecutan de man-

era independiente su propio sistema operativo y dentro de él los servicios que quieran ofrecer, haciendo un uso común de manera compartida y aislada sin ser conscientes del hardware subyacente

##### **1.1 Sistemas operativos invitados**

---

Sobre una aplicación para virtualización no hace uso de hipervisor u otra capa de virtualización que corre sobre la instancia de un sistema operativo sistema operativo host se permite la ejecución de servidores virtuales con sistemas operativos independientes.

##### **1.2 Emulacion**

---

Un emulador que replica una arquitectura hardware al completo procesador, juego de instrucciones, periféricos hardware-permite que se ejecuten sobre él máquinas virtuales.

##### **1.3 Virtualizacion Completa**

---

También llamada nativa. La capa de virtualización, un hipervisor, media entre los sistemas invitados y el anfitrión, la cual incluye código que emula el hardware subyacente –si es necesario- para las máquinas virtuales, por lo que es posible ejecutar cualquier sistema operativo sin modificar, siempre que soporte el hardware subyacente.

##### **1.4 Virtualizacion a nivel del sistema operativo**

---

Virtualiza los servidores sobre el propio sistema operativo, sin introducir una capa intermedia de virtualización.

##### **1.5 Virtualizacion a nivel del kernel**

---

Convierte el núcleo Linux en hipervisor utilizando un módulo, el cual permite ejecutar máquinas virtuales y otras instancias de sistemas operativos en el espacio de usuario del núcleo Linux anfitrión

## 2. Virtualización de recursos

---

En este segundo caso el recurso que se abstrae es un recurso individual de un computador, como puede ser la conexión a red, el almacenamiento principal y secundario, o la entrada y salida.

### 2.1 Encapsulamiento

---

Se trata de la ocultación de la complejidad y características del recurso creando una interfaz simplificada. Es el caso más simple de virtualización de recursos, como se puede ver.

### 2.2 Memoria virtual

---

Permite hacer creer al sistema que dispone de mayor cantidad de memoria principal y que se compone de segmentos contiguos. Como sabemos, es usada en todos los sistemas operativos modernos.

### 2.3 Virtualización de almacenamiento

---

Abstracción completa del almacenamiento lógico sobre el físico (disco y almacenamiento son el recurso abstraído). Es completamente independiente de los dispositivos hardware.

### 2.3 Virtualización de red

---

La virtualización de red consiste en la creación de un espacio de direcciones de red virtualizado dentro de otro o entre subredes. Es fácil ver que el recurso abstraído es la propia red. Ejemplos bien conocidos de virtualización de red son OpenVPN y OpenSwarm, que permiten crear VPNs.

## 3. Virtualización de aplicaciones

---

Las aplicaciones son ejecutadas encapsuladas sobre el sistema operativo de manera que aunque creen que interactúan con él de la manera habitual, en realidad no lo hacen, sino que lo hacen bien con una máquina virtual de aplicación o con algún software de virtualización.

## 4. Virtualización de escritorio

---

Consiste en la manipulación de forma remota del escritorio de usuario (aplicaciones, archivos, datos), que se encuentra separado de la máquina física, almacenado en un servidor central remoto en lugar de en el disco duro del computador local.

### B). Cuales son las Ventajas de la Virtualización

- Hardware de los Servidores Infrautilizado
- Se agota el Espacio en los Data Centers
- Demanda de una mejor Eficiencia Energética

## ii. Contenedores

### ¿Que son los contenedores?

Es una pregunta básica, pero necesaria. Podemos pensar en un contenedor como un servidor que arranca desde una imagen estática predefinida con un sistema operativo con un kernel Linux y con las librerías y recursos mínimos necesarios de CPU, memoria, almacenamiento, etc. Realmente, un contenedor consiste en el empaquetado de una aplicación para que pueda correr en cualquier sistema abstrayéndose de la plataforma sobre la que está corriendo.

### ¿Para qué se pueden usar los contenedores?

Con una aplicación empaquetada con versiones de sistema, librerías y demás recursos óptimos para su ejecución, evitamos problemas de incompatibilidades. Por ejemplo, **si los contenedores son independientes de las características del servidor donde se alojan**, evitamos problemas a la hora de que actualicemos este. Otro ejemplo es que un contenedor puede ejecutarse en cualquier servidor Linux, Mi-

Microsoft e incluso en MacOS nos facilita las migraciones entre máquinas. Las aplicaciones que pueden ejecutarse en un contenedor son múltiples y variadas, **desde un apache a una BBDD Mongo.**

### ¿Qué ventajas nos ofrecen los contenedores sobre un servidor físico o virtual?

Al principio del texto hablábamos de los cuatro pilares que sostienen el éxito de los servicios que nos ofrecen desde la nube pública y que también son aplicables cuando trabajamos con contenedores. Con el uso de orquestadores de contenedores tipo Kubernetes, Openshift, Mesos o un largo etcétera, podemos asegurar que **nuestra aplicación va a estar siempre disponible para su ejecución.** Podremos aumentar o disminuir el número de contenedores en función de nuestras necesidades fácilmente y asegurar que, en caso de fallo del contenedor por cualquier causa, este podrá arrancar automáticamente de manera instantánea.

Los contenedores están diseñados **para que tengan una vida corta pero útil: arrancan, ejecutan su función y mueren.** Debido a que se trata de sistemas muy livianos, los contenedores normalmente tardan pocos segundos en arrancar. Por el hecho de ser tan efímeros, están enfocados a ejecutar la aplicación o aplicaciones para la que fueron creados y una vez finalizada, se elimina dicho contenedor. De esta manera, mientras no necesitemos ejecutar una aplicación en concreto, estamos dejando libres los recursos del servidor anfitrión para poder ser utilizados por otros contenedores. **De esta manera no hace falta mantener un servidor encendido 24 horas** consumiendo energía y recursos si solo va hacer, por ejemplo, una inserción de una tabla en una base de datos en un proceso batch a final de mes.

**¿Puedo usar un contenedor como un**

### **servidor normal?**

Podemos acceder a un contenedor a través de una shell como si de un servidor normal se tratase. También podemos ejecutar comandos contra un contenedor que esté corriendo, pero tenemos que tener en cuenta que, como hemos indicado antes, un contenedor arranca de una imagen estática predefinida. Es decir, **salvo que lo indiquemos a propósito**, todos los cambios que hagamos en nuestro contenedor, los perderemos cuando lo matemos. Si arrancamos un nuevo contenedor con la misma imagen, tendremos la misma configuración que tenía antes de hacer ningún cambio.

### ¿Qué pasa con mis cambios realizados sobre la imagen del contenedor?

Cuando tenemos un contenedor corriendo, podemos hacer cualquier cambio de configuración sobre su imagen inicial. En el caso de que quisiéramos conservar los cambios que hemos aplicado, podemos salvarlos en una imagen nueva. Esta nueva imagen, a su vez, estaría disponible para ser utilizada por otros contenedores y así sucesivamente.

## A). Virtualización de Contenedores

La virtualización basada en contenedores, también llamada virtualización del sistema operativo, es una aproximación a la virtualización en la cual la capa de virtualización se ejecuta como una aplicación en el sistema operativo (OS). En este enfoque, el kernel del sistema operativo se ejecuta sobre el nodo de hardware con varias máquinas virtuales (VM) invitadas aisladas, que están instaladas sobre el mismo. Los huéspedes aislados se denominan contenedores.

Con la virtualización basada en contenedores, no existe la sobrecarga asociada con tener a cada huésped ejecutando un sis-

tema operativo completamente instalado. Este enfoque también puede mejorar el rendimiento porque hay un solo sistema operativo encargándose de los avisos de hardware. Una desventaja de la virtualización basada en contenedores, sin embargo, es que cada invitado debe utilizar el mismo sistema operativo que utiliza el host.

Por lo general, los entornos corporativos evitan la virtualización basada en contenedores, prefiriendo los hipervisores y la opción de tener muchos sistemas operativos. Un entorno virtual basado en contenedor, sin embargo, es una opción ideal para los proveedores de alojamiento que necesitan una manera eficiente y segura para ofrecer sistemas operativos para que los clientes ejecuten sus servicios en ellos.

## **B). Diferencia entre Virtualización Clásica y Virtualización de Contenedores**

### **- Virtualización clásica**

La virtualización tal y como se entiende habitualmente supone el despliegue de múltiples máquinas software completas dentro de un mismo hardware. Dependiendo de cómo se hace el despliegue existen dos tipos de virtualización, hipervisores de tipo 1 o instalado directamente como sistema operativo, e hipervisores de tipo 2 o instalados sobre un sistema operativo comercial; como se comentaron en Mi SCADA en las nubes.

La virtualización completa de las máquinas permite que unas sean independientes de las otras, de manera que se pueden gestionar como si fuesen máquinas hardware separadas, pero favoreciendo la ampliación de recursos en caso de ser necesario.

### **- Virtualización mediante contenedores**

El último avance de la virtualización es la utilización de contenedores. Con esta tecnología no se virtualiza el sistema entero, sino que, partiendo de una imagen de base, se registran los cambios realizados

tanto por instalaciones como por desinstalaciones, de aplicaciones y servicios. De esta manera los ficheros de las imágenes de las máquinas son mucho menores y las necesidades de espacio se reducen considerablemente.

## **-Comparación entre virtualización tradicional y con contenedores**

Los contenedores están aislados unos de otros, pero comparten un mismo sistema operativo, librerías y binarios. Esto hace que el despliegue sea mucho más rápido que una instalación nueva, así como los reinicios y las migraciones, pero tiene el problema de que una vulnerabilidad en una máquina anfitrión puede afectar también al resto, al compartir la base del sistema operativo.

## **C).Ventajas y Desventajas**

La principal motivación para virtualizar un sistema SCADA es reducir hardware e infraestructura, y con ello los gastos; pero también existen otras razones.

La virtualización de los sistemas de control permite integrarlos en el entorno corporativo y con ello estrechar los lazos entre diferentes departamentos y compartir responsabilidades y toma de decisiones, mejorando la colaboración y la integración de medidas de seguridad.

Como la virtualización se utiliza para agrupar máquinas en un mismo hardware, que son posteriormente accedidas de forma remota, esta característica permite mejorar el trabajo del personal que debe acceder a información con diferentes niveles de seguridad, utilizando diferentes máquinas virtuales correctamente segmentadas en lugar de diferentes equipos separados por air-gap.

**Ventajas:**

Reducción de hardware y costes operacionales  
 Rápida recuperación frente a desastres a snapshots y clones  
 Pruebas seguras arquitecturas de seguridad  
 Mejora del desarrollo y aseguramiento de la calidad

**Desventajas:**

Dificultad para encontrar el origen del problema.  
 Posibilidad de perder muchos sistemas en un servidor  
 Incremento de la superficie de ataque ya que se añade el hipervisor  
 Posibles problemas con interfaces físicos, como puertos USB, RS.232 etc.  
 Rendimiento

**iii. Contenedores con Docker**

Docker es una plataforma de software que le permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución. Con Docker, puede implementar y ajustar la escala de aplicaciones rápidamente en cualquier entorno con la certeza de saber que su código se ejecutará.

La ejecución de Docker en AWS les ofrece a los desarrolladores y administradores una manera muy confiable y económica de crear, enviar y ejecutar aplicaciones distribuidas en cualquier escala. AWS es compatible con ambos modelos de licencia de Docker: Docker Community Edition (CE) de código abierto y Docker Enterprise Edition (EE) basada en suscripción.

Docker, me permite meter en un contene-

dor ("una caja", algo auto contenido, cerrado) todas aquellas cosas que mi aplicación necesita para ser ejecutada (java, Maven, tomcat...) y la propia aplicación. Así yo me puedo llevar ese contenedor a cualquier máquina que tenga instalado Docker y ejecutar la aplicación sin tener que hacer nada más, ni preocuparme de qué versiones de software tiene instalada esa máquina, de si tiene los elementos necesarios para que funcione mi aplicación, de si son compatibles.

**A). Orquesta de Aplicación**

Existen múltiples definiciones sobre el concepto de orquestación de aplicaciones pero de un modo simple, podemos definir la orquestación de servicios o aplicaciones como el uso de la automatización para la creación y composición de la arquitectura, herramientas y procesos utilizados por operadores humanos para entregar un servicio.

La orquestación aprovecha tareas automatizadas y procesos predefinidos para permitir la creación de infraestructura complejas y para conseguir el aprovechamiento de los recursos de forma óptima y automatizada. Podemos considerar, a modo de analogía, el concepto de orquestación como un proceso y la automatización como una tarea.

De este modo, el objetivo principal de la orquestación consiste en la automatización de procesos orientados al despliegue y ciclo de vida de las aplicaciones o servicios. Y la automatización de procesos en los despliegues software se basan en el uso de algún tipo de software que facilite la

**B). Docker y otros container: más allá de la virtualización**

En un mundo donde cada vez es más común el uso de servicios informáticos en la nube y cuyos principios básicos para explicar su éxito radican en la alta disponibilidad, el diseño de tolerancia a fallos, el escalado y la elasticidad, parece que se hace obligatorio que hablemos sobre los contenedores. En este post nos referimos

en concreto al proyecto “Docker”.

Cuando pensábamos que ya habíamos alcanzado el máximo ahorro de recursos con la virtualización de todo el hardware posible de nuestra infraestructura (servidores, redes, cabinas de discos, etc.), desde hace poco tiempo se ha extendido el uso de los llamados contenedores. La idea de esta post es repasar los conceptos básicos que rodean al mundo de los contenedores y dar una visión general sobre esta herramienta.

### **C). Contenedor Docker, la Tecnología de Contenedores a Mano**

Para hablar de futuro, es necesario observar el camino recorrido y los avances tecnológicos no son ajenos a ellos. Si hablamos de virtualización 3.0, se podría decir que inició desde el uso de los primeros mainframes los cuales permitían que varios usuarios operaran al mismo tiempo a través de los “terminales tontos”.

### **D). ¿Son Seguros los Contenedores**

Podemos pensar en un contenedor como un servidor que arranca desde una imagen estática predefinida con un sistema operativo con un kernel Linux y con las librerías y recursos mínimos necesarios de CPU, memoria, almacenamiento, etc. Realmente, un contenedor consiste en el empaquetado de una aplicación para que pueda correr en cualquier sistema abstrayéndose de la plataforma sobre la que está corriendo.

### **iv. Hypervisores Bare Metal**

Hypervisores de tipo 1 (También llamados nativos, unhosted o bare-metal) Un hypervisor bare-metal no funciona bajo un sistema operativo instalado sino que tiene acceso directo sobre los recursos hardware; el hypervisor se carga antes que ninguno de los sistemas operativos invitados, y todos los accesos directos a hardware son controlados por él. Aunque esta es la aproximación clásica y más antigua de la

virtualización por hardware, actualmente las soluciones más potentes de la mayoría de fabricantes usa este enfoque. Nota: es muy frecuente que a los hipervisores en general se les aplique el término VMM (Monitores de máquina virtual), mientras que el término “Hypervisor” se reserva para los hipervisores de tipo 1.

### **A). Tipos de Hypervisores**

#### **1) Monolítico**

Son hipervisores que emulan hardware para sus máquinas virtuales. Esta aproximación, usada por productos como VMWare ESX, obliga a usar una gran cantidad de código que se interpone entre los recursos físicos reales y las operaciones de acceso a ellos efectuadas por las máquinas virtuales.

El proceso que sigue una llamada a hardware en un sistema virtualizado usando un hypervisor de tipo monolítico es:

- 1.- El hardware emulado debe interceptar la llamada.
- 2.- El VMM redirige estas llamadas hacia los drivers de dispositivo que operan dentro del hypervisor, lo cual requiere de numerosos cambios de contexto en el código de la llamada.
- 3.- Los drivers del hypervisor enrutan la llamada hacia el dispositivo físico.

Este funcionamiento obliga a desarrollar drivers específicos para el hypervisor de cada componente hardware.

#### **2) De MicroKernel**

En esta aproximación el hypervisor se reduce a una capa de software muy sencilla, cuya única funcionalidad es la de particionar el sistema físico entre los diversos sistemas virtualizados.

Con esta manera de funcionar los hipervisores de microkernel no requieren de drivers específicos para acceder al hardware.



En el caso de Hyper-V, el acceso a los recursos físicos se hace desde la partición primaria, usando los drivers nativos de Windows Server 2008 R2.

En las particiones hija se utilizan drivers sintéticos, que son simplemente enlaces a los drivers reales ubicados en la partición primaria.

De esta manera los hipervisores de microkernel no sólo aumentan el rendimiento al reducir el código intermedio y el número de cambios de contexto necesarios, sino que también aumentan la estabilidad de los sistemas, al haber menos componentes, y sobre todo la seguridad, al reducir la superficie de ataque del hipervisor.

## B). Ventajas y Desventajas

### Ventajas:

Obtendremos un mejor rendimiento, escalabilidad y estabilidad.

### Desventajas:

En este tipo de tecnología de virtualización el hardware soportado es más limitado ya que normalmente es construido con un conjunto limitado de drivers.

## C). Productos Comerciales

Entre los hipervisores de este tipo encontramos: VMware ESX o ESXi, Microsoft Hyper-V, Citrix XenServer u Oracle VM.

### VMware ESX y ESXi

VMware es el fabricante con la tecnología de virtualización más madura del mercado. Ofrece características avanzadas y escalabilidad. En contra tiene los altos costes de licenciamiento.

### Microsoft Hyper-V

Hyper-V se ha convertido en un serio competidor de VMware ESX (ESXi). En contra tiene que no dispone de ciertas características avanzadas disponibles en los productos de VMware. De todos modos, como no podía ser de otra forma, se integra perfectamente con los productos Windows. Para aquellos que no necesitan funcionalidades avanzadas, puede ser un producto perfecto para llevar a cabo su proyecto de

virtualización.

### Citrix XenServer

Citrix también tiene una plataforma de virtualización basada en el proyecto open source Xen. El hipervisor es gratis, pero de igual modo a como pasa con VMware ESXi, no dispone de características avanzadas. Éstas se obtienen a partir de licencias que ofrecen gestión avanzada, automatización y alta disponibilidad.

### OracleVM

De igual modo a Citrix, Oracle ha desarrollado su hipervisor a partir del proyecto Xen. El producto de Oracle no presenta funcionalidades avanzadas que podemos encontrar en otros hipervisores bare-metal. Además su ciclo de desarrollo es lento y limitado, por lo que no puede competir con los productos de VMware, Microsoft o Citrix. Sin embargo, como es lógico, es un producto que se adapta perfectamente a los productos de Oracle.

## v. Maquinas virtuales VS Contenedores

## IV. CONCLUSIONES

Ambas tecnologías ofrecen ventajas distintas:

La virtualización viene con una plétora de herramientas probadas a lo largo del tiempo, plataformas de gestión y orquestación, sondas virtuales, soluciones de infraestructura virtual hiperconvertidas y mucho más. La portabilidad y la interoperabilidad son las características que destacan frente a los contenedores.

Los contenedores ofrecen una mayor eficiencia de recursos y agilidad de servicio. Aunque no parezca mucho, abre la puerta a un modelo de microservicios que puede escalar más rápido y de manera más eficiente. Los contenedores de papel se ajustan más a las iniciativas de NFV/SDN y la

industria se ha dado cuenta de que Kubernetes es uno de los proyectos de código abierto de más rápido crecimiento hasta la fecha.

#### REFERENCES

1. <http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/23/21>
2. <https://programarfacil.com/blog/que-es-un-orm/>
3. <https://www.beeva.com/beeva-view/tecnologia/mas-alla-de-la-virtualizacion-contenedores/>
4. <https://searchdatacenter.techtarget.com/es/definicion/virtualizacion-basada-en-contenedores-virtualizacion-a-nivel-de-sistema-operativo>
5. <https://www.incibe-cert.es/blog/asegurando-virtualizacion-tus-sistemas-control>
6. <http://www.datakeeper.es/?p=716>