



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

**SOFTWARE DE ADMINISTRACIÓN DE RECURSOS EN UN AMBIENTE VIRTUALIZADO DE  
LOS SERVIDORES QUE UTILIZA LA DIVISIÓN DE EDUCACIÓN A DISTANCIA (DEDEV) DE  
LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

**Gary Stephen Girón Molina**  
**José Rodolfo Portillo Samayoa**  
**Abner Lorenzo López Chávez**

Asesorado por el Ing. Neftalí de Jesús Calderón Méndez

Guatemala, octubre de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**SOFTWARE DE ADMINISTRACIÓN DE RECURSOS EN UN AMBIENTE VIRTUALIZADO DE  
LOS SERVIDORES QUE UTILIZA LA DIVISIÓN DE EDUCACIÓN A DISTANCIA (DEDEV) DE  
LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR

**GARY STEPHEN GIRÓN MOLINA**

**JOSÉ RODOLFO PORTILLO SAMAYOA**

**ABNER LORENZO LÓPEZ CHÁVEZ**

ASESORADO POR EL ING. NEFTALÍ DE JESÚS CALDERÓN MÉNDEZ

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, OCTUBRE DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANA	Inga. Aurelia Anabela Cordoba Estrada
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Ing. César Rolando Batz Saquimux
EXAMINADOR	Ing. José Manuel Ruiz Juárez
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**Gary Stephen Girón Molina**

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANA	Inga. Aurelia Anabela Cordoba Estrada
EXAMINADOR	Ing. Herman Igor Veliz Linares
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADORA	Inga. Devora Emperatriz Meza Orellana
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**José Rodolfo Portillo Samayoa**

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANA	Inga. Aurelia Anabela Cordoba Estrada
EXAMINADOR	Ing. José Alfredo Gonzalez Gonzalez
EXAMINADOR	Ing. Marlon Francisco Orellana López
EXAMINADOR	Ing. Nefalí de Jesús Calderón Méndez
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**Abner Lorenzo López Chávez**

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**SOFTWARE DE ADMINISTRACIÓN DE RECURSOS EN UN AMBIENTE VIRTUALIZADO DE LOS SERVIDORES QUE UTILIZA LA DIVISIÓN DE EDUCACIÓN A DISTANCIA (DEDEV) DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 14 de agosto de 2017.

**Gary Stephen Girón Molina**

**José Rodolfo Portillo Samayoa**

**Abner Lorenzo López Chávez**

Guatemala, 30 de agosto de 2021

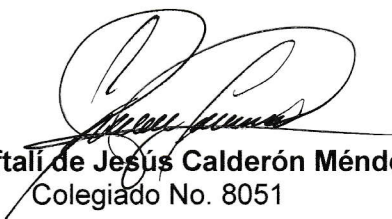
Ingeniero  
**Carlos Alfredo Azurdia**  
Coordinador de Privados y Trabajos de Tesis  
Escuela de Ingeniería en Ciencias y Sistemas  
Facultad de Ingeniería - USAC

Respetable Ingeniero Azurdia:

Por este medio hago de su conocimiento que en mi rol de asesor del trabajo de investigación realizado por el estudiante **GARY STEPHEN GIRÓN MOLINA** con carné 201403997 y CUI 2954 72855 0101, **JOSÉ RODOLFO PORTILLO SAMAYOA** con carné 201313720 y CUI 2317 38633 0108, **ABNER LORENZO LÓPEZ CHÁVEZ** con carné 201114705 y CUI 2078 59426 0101 titulado **“SOFTWARE DE ADMINISTRACIÓN DE RECURSOS EN UN AMBIENTE VIRTUALIZADO DE LOS SERVIDORES QUE UTILIZA LA DIVISIÓN DE EDUCACIÓN A DISTANCIA (DEDEV) DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, lo he revisado y luego de corroborar que el mismo se encuentra concluido y que cumple con los objetivos propuestos en el respectivo protocolo, procedo a la aprobación respectiva.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



**Ing. Neftalí de Jesús Calderón Méndez**  
Colegiado No. 8051



Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 3 de septiembre del 2021

Ingeniero  
**Carlos Gustavo Alonzo**  
**Director de la Escuela de Ingeniería**  
**En Ciencias y Sistemas**

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **GARY STEPHEN GIRÓN MOLINA** con carné **201403997** y CUI **2954 72855 0101**, **JOSÉ RODOLFO PORTILLO SAMAYOA** con carné **201313720** y CUI **2317 38633 0108** y **ABNER LORENZO LÓPEZ CHÁVEZ** con carné **201114705** y CUI **2078 59426 0101**, titulado: “**SOFTWARE DE ADMINISTRACIÓN DE RECURSOS EN UN AMBIENTE VIRTUALIZADO DE LOS SERVIDORES QUE UTILIZA LA DIVISIÓN DE EDUCACIÓN A DISTANCIA (DEDEV) DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**”, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN  
CIENCIAS Y SISTEMAS

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“SOFTWARE DE ADMINISTRACIÓN DE RECURSOS EN UN AMBIENTE VIRTUALIZADO DE LOS SERVIDORES QUE UTILIZA LA DIVISIÓN DE EDUCACIÓN A DISTANCIA (DEDEV) DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, realizado por los estudiantes, GARY STEPHEN GIRÓN MOLINA, JOSÉ RODOLFO PORTILLO SAMAYOA y ABNER LORENZO LÓPEZ CHÁVEZ aprueba el presente trabajo y solicita la autorización del mismo.*

**“ID Y ENSEÑAD A TODOS”**

A handwritten signature in black ink is written over an official circular stamp. The stamp contains the text 'UNIVERSIDAD DE SAN CARLOS DE GUATEMALA' and 'DIRECCION DE INGENIERIA EN CIENCIAS Y SISTEMAS'.

MSC. CARLOS GUSTAVO AIZONZO

**Director**

**Escuela de Ingeniería en Ciencias y Sistemas**

Guatemala, 12 de octubre de 2021




**USAC**  
TRICENTENARIA  
Universidad de San Carlos de Guatemala


Decanato  
Facultad de Ingeniería  
24189101- 24189102  
secretariadecanato@ingenieria.usac.edu.gt

DTG. 517.2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **SOFTWARE DE ADMINISTRACIÓN DE RECURSOS EN UN AMBIENTE VIRTUALIZADO DE LOS SERVIDORES QUE UTILIZA LA DIVISIÓN DE EDUCACIÓN A DISTANCIA (DEDEV) DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por los estudiantes universitarios: **Gary Stephen Girón Molina, José Rodolfo Portillo Samayoa y Abner Lorenzo López Chávez**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

  
Inga. Anabela Cordova Estrada  
Decana



Guatemala, octubre de 2021

AACE/asga

## **ACTO QUE DEDICO A:**

### **Jehová Dios**

Por permitirme la vida, me ha guiado en los pasos que he dado y me ha provisto de la fortaleza y determinación para llegar al final de la meta.

### **Mi madre**

Erika Yasmina Molina González, por ser quien siempre confió en mí y mi esfuerzo, por su apoyo y amor incondicional que siempre fue una motivación e inspiración para seguir adelante.

### **Mi abuela**

Amelia Morales González, quien con su cuidado, amor y apoyo en vida fue determinante para trazarme mis objetivos.

### **Mi tío**

José Enrique Morales González, quien siempre confió en mí y recuerdo que estuvo acompañándonos en momentos importantes de mi trayecto.

**Gary Stephen Girón Molina**

## **ACTO QUE DEDICO A:**

- Dios** Por ser la luz que guio mi camino y la fortaleza para poder culminar con éxito este logro.
- Mi madre** Sandra Marleny Portillo Samayoa, por ser ese apoyo incondicional durante mi carrera y por el esfuerzo entregado en este proceso.
- Mi abuela** Milagro Samayoa, por ser siempre la persona que me ponía en sus oraciones para que Dios me iluminara y me protegiera.
- Mis hermanos** Hugo Leonel Portillo Samayoa y Jospheh Ivan Schumann Portillo, que con sus palabras de aliento me animaban a seguir adelante.
- Mi novia** Ana Elisa Chew Aldana, por ser esa persona que me impulsó a nunca rendirme con su ejemplo y que creyó en mí hasta el final.

**José Rodolfo Portillo Samayoa**

## **ACTO QUE DEDICO A:**

- Dios** Por brindarme el conocimiento, la sabiduría y la oportunidad de alcanzar esta meta.
- Mis padres** Daniel Andrés López Tax y Dalila Betsabé Chávez Vásquez por ser el pilar fundamental de todos los triunfos en mi vida.
- Mis hermanos** Daniel Andrés López Chávez y Samara Betsabé López Chávez por sus buenos ejemplos y el apoyo incondicional que me brindan en todo momento.
- Toda mi familia** Por estar siempre pendientes de mí y animarme continuar hasta alcanzar este triunfo

**Abner Lorenzo López Chávez**

## **AGRADECIMIENTOS A:**

**Universidad de San  
Carlos de Guatemala**

Por permitirme ser parte de los estudiantes que aspiran ser egresados de este magno centro de estudios.

**Facultad de Ingeniería**

Por ser aquella que me brindó los conocimientos que hoy puedo poner en práctica, donde pasé varios años de mi vida y ahora culminan en una meta cumplida.

**Congreso de Estudiantes  
de Ciencias y Sistemas**

Por brindarme la oportunidad de formar parte de la planilla que llevo a cabo este evento como parte de mis prácticas, y me dio las herramientas para desenvolverme en diferentes ámbitos que hoy han definido quién soy como profesional.

**Mis amigos**

Porque cada uno ha sido fundamental para llegar al punto en el que estoy, han formado parte de la persona que soy y los logros que he llegado a conseguir.

**Gary Stephen Girón Molina**

## **AGRADECIMIENTOS A:**

**Universidad de San  
Carlos de Guatemala**

Por abrirme las puertas y permitir que pudiera aprender de los mejores catedráticos.

**Facultad de Ingeniería**

Por ser mi segundo hogar y por brindarme las herramientas necesarias para formarme como persona y como profesional.

**Departamento  
de Matemáticas**

Por brindarme el espacio y permitir que pudiera conocer personas excepcionales y dejarme cumplir el sueño de ser auxiliar de la Facultad de Ingeniería.

**Mis amigos**

Porque cada uno tuvo un papel muy importante tanto en mi carrera como en mi vida personal, cada uno de ustedes sabe que aportó y que hizo en el tiempo que nos conocimos, ya que fueron una importante influencia para poder seguir adelante en esta carrera que dejó lágrimas y risas pero que al final se logró.

**José Rodolfo Portillo Samayoa**

## **AGRADECIMIENTOS A:**

**Universidad de San  
Carlos de Guatemala**

Por brindarme la oportunidad de pertenecer a esta casa de estudios.

**Facultad de Ingeniería**

Por todo el conocimiento y recursos brindados para mi formación profesional como Ingeniero en Ciencias y Sistemas.

**Departamento  
de Matemáticas**

Por brindarme la oportunidad de pertenecer a este gran departamento y todas las valiosas amistades que conocí en este lugar.

**Mis amigos**

Por sus ejemplos, ánimos, conocimiento, apoyo y todas las demás cosas que ayudaron a finalizar esta carrera.

**Abner Lorenzo López Chávez**





1.2.3.	Desventajas.....	6
1.3.	Máquina virtual.....	6
1.3.1.	Uso.....	7
1.3.2.	Tipos de máquinas virtuales.....	7
1.3.2.1.	Máquinas virtuales de procesos.....	7
1.3.2.2.	Máquinas virtuales de sistema.....	8
1.3.3.	Ventajas.....	8
1.3.4.	Desventajas.....	8
1.4.	Gestor de contenidos.....	9
1.4.1.	Uso.....	9
1.4.2.	Ventajas.....	10
1.4.3.	Desventajas.....	10
1.5.	Contenedores.....	11
1.5.1.	Uso.....	11
1.5.2.	Tipos.....	12
1.5.2.1.	rkt.....	12
1.5.2.2.	LXC.....	13
1.5.2.3.	LXD.....	14
1.5.2.4.	Docker.....	14
1.5.3.	Ventajas.....	15
1.5.4.	Desventajas.....	16
1.6.	<i>Cloud Computing</i> .....	16
1.6.1.	Uso.....	17
1.6.2.	Tipos o modelos de servicio en la nube.....	18
1.6.2.1.	Software como servicio (SaaS).....	19
1.6.2.2.	Plataforma como servicio (PaaS).....	20
1.6.2.3.	Infraestructura como servicio (IaaS).....	21
1.6.3.	Ventajas.....	22
1.6.4.	Desventajas.....	23

1.7.	Educación en línea .....	23
1.7.1.	Uso .....	23
1.7.2.	Tipos.....	24
1.7.3.	Ventajas.....	24
1.7.4.	Desventajas .....	25
2.	ANÁLISIS Y COMPARACIÓN DE SOLUCIONES .....	27
2.1.	Máquinas virtuales por cada unidad académica en el servidor (máquinas virtuales como instancias) .....	28
2.1.1.	Puntos a favor de la solución.....	29
2.1.2.	Puntos en contra de la solución.....	30
2.2.	Instancias de contenedores necesarias por cada unidad académica en el servidor (contenedores como instancias) .....	30
2.2.1.	Puntos a favor de la solución.....	32
2.2.2.	Puntos en contra de la solución.....	32
2.3.	Instalación de un entorno de virtualización que gestione los contenedores para cada unidad académica en el servidor (entorno de virtualización para contenedores).....	33
2.3.1.	Puntos a favor de la solución.....	34
2.3.1.1.	Integridad de los contenedores de <i>software</i> .....	35
2.3.1.2.	Versatilidad de los contenedores de <i>software</i> .....	35
2.3.2.	Puntos en contra de la solución.....	35
2.3.2.1.	Administración de recursos.....	36
2.4.	Instalación de un entorno de virtualización que gestione las máquinas virtuales, las cuales será una por unidad académica, dentro de las cuales se tendrán contenedores por cada servicio necesario para el sistema	

correspondiente (entorno de virtualización para máquinas virtuales).....	36
2.4.1. Puntos a favor de la solución .....	38
2.4.1.1. Modularidad.....	39
2.4.1.2. Administración de recursos .....	39
2.4.1.3. Versatilidad de la solución.....	40
2.4.1.4. Manejo de errores .....	40
2.4.2. Puntos en contra de la solución .....	41
2.4.2.1. Desperdicio de recursos.....	41
2.4.2.2. Mayor tiempo de implementación.....	41
2.5. Requerimientos de solución .....	42
3. SOLUCIÓN PROPUESTA .....	47
3.1. Plataforma para gestión y creación de contenedores .....	47
3.1.1. ¿Por qué utilizar contenedores?.....	48
3.1.1.1. Definición de Docker .....	49
3.1.1.1.1. Uso de Docker .....	50
3.1.2. Hipervisor .....	51
3.1.2.1. Definición y funcionamiento de Proxmox Virtual Environment como hipervisor.....	52
3.1.2.1.1. Uso de Proxmox Virtual Environment Virtualización mediante KVM.....	53
3.1.3. Plataforma de gestión de virtualización.....	54
3.1.3.1. Definición y funcionamiento de Proxmox Virtual Environment como	

	plataforma de gestión de virtualización .....	55
3.1.4.	Sistema operativo recomendado .....	57
3.1.5.	Instalación de Proxmox utilizando Proxmox VE Installer .....	58
3.1.6.	Instalación de Proxmox sobre Debian Buster .....	59
3.2.	Impacto social y de desarrollo a nivel nacional de la propuesta de solución .....	59
4.	DISEÑO Y PROPUESTA PARA IMPLEMENTACIÓN DE SEGURIDAD .....	63
4.1.	Diseño de seguridad.....	63
4.1.1.	Seguridad en la Nube .....	65
4.1.2.	Seguridad en máquinas virtuales.....	68
4.1.3.	Seguridad en contenedores.....	70
4.2.	Diseño de la arquitectura.....	71
4.2.1.	Nivel Externo - MSSP.....	72
4.2.2.	Nivel Interno.....	76
	CONCLUSIONES .....	81
	RECOMENDACIONES .....	83
	BIBLIOGRAFÍA.....	85
	APÉNDICE.....	89



## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1.	Disponibilidad de acceso.....	19
2.	Plataforma de servicio.....	21
3.	Infraestructura de servicio.....	22
4.	Estructura.....	29
5.	Servidor.....	31
6.	Entorno de virtualización para contenedores.....	34
7.	Entorno de virtualización para máquinas virtuales.....	38
8.	Arquitectura, parte 1.....	74
9.	Arquitectura, parte 2.....	77
10.	Arquitectura propuesta.....	79





## LISTA DE SÍMBOLOS

<b>Símbolo</b>	<b>Significado</b>
<b>MB</b>	MegaByte
<b>GB</b>	GigaByte



## GLOSARIO

<b>Ágil</b>	Es un enfoque para el desarrollo de software que busca distribuir de forma permanente los sistemas en funcionamiento diseñados con iteraciones rápidas.
<b>AGPL</b>	Licencia.
<b>Alojamiento</b>	Es el servicio que provee a los usuarios de un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido.
<b>Amazon Web Services</b>	Es la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 200 servicios integrales de centros de datos a nivel global.
<b>Ambiente</b>	Espacio que rodea o está en el entorno de trabajo que provee los recursos con los cuales realizar las actividades.
<b>AMD</b>	<i>Advanced Micro Devices</i> (Competencia de intel).
<b>amd64</b>	Arquitectura de procesador de 64 bits que fue desarrollado por <i>Advanced Micro Devices</i> (AMD).

<b>Apache</b>	Es un servicio de páginas web HTTP de código abierto que sirve para colocar varias plataformas como Unix, BSD, GNU/Linux, Windows, Macintosh entre otros que implementan dicho protocolo.
<b>API</b>	Interface de Programa de aplicación (Application Prog Interface).
<b>Aplicaciones</b>	Es un tipo de software de computadora diseñado para realizar un grupo de funciones.
<b>Ataque cibernético</b>	Es una acción delictiva y malintencionada que se realiza para acceder a información privada, para apropiarse de ella u obtener un beneficio.
<b>Azure</b>	Es una nube pública de pago por uso que permite compilar, implementar y administrar rápidamente aplicaciones en una red global de <i>datacenters</i> de Microsoft.
<b>Capa intermedia</b>	Se considera así al espacio del proceso que se lleva de forma interna dentro del sistema, el cual no conlleva la entrada o salida de la información sino únicamente su desarrollo.
<b>CLI</b>	Interfaz de línea de comandos.
<b>Cloud Computing</b>	La computación en la nube es una tecnología que permite acceso remoto a softwares, almacenamiento

de archivos y procesamiento de datos por medio de internet, siendo así, una alternativa a la ejecución en una computadora personal o servidor local.

***Código fuente***

Conjunto de líneas de textos, que son las directrices que debe seguir la computadora para realizar dicho programa.

***Código máquina***

Se trata del lenguaje que pueden ejecutar los ordenadores y otros sistemas de computación, el cual está basado en un sistema binario.

***Compatible***

La capacidad que tiene cierto sistema para funcionar simultáneamente con otros sistemas permitiendo o mejorando el funcionamiento del conjunto.

***CPU***

Unidad de procesamiento central (*Central processing unit*).

***Debian***

Es un sistema operativo y una distribución de software libre que se mantiene y actualiza por la colaboración de muchos usuarios.

***DEDEV***

División de educación a distancia en entornos virtuales.

***Desarrollo***

Construcción de software mediante su descripción.

<b>DevOps</b>	Es una combinación de los términos ingleses: desarrollo y operaciones, que designa la unión de personas, procesos y tecnología para ofrecer valor a los clientes de forma constante.
<b>Disco Duro</b>	Es un dispositivo de almacenamiento de datos no volátil y emplea un sistema de grabación magnético para guardar los datos digitales.
<b>Docker</b>	Herramienta de contenedorización de código abierto.
<b>Ejecución</b>	Acción de poner en marcha cierto proyecto o aplicación.
<b>Entorno</b>	Se trata de las herramientas, equipos y conforman la red de trabajo en la que se encuentra una terminal.
<b>Equipos físicos</b>	Se refiere al equipo de cómputo que se encuentra presente en el espacio de trabajo.
<b>Equipos virtuales</b>	Se refiere al equipo de cómputo que se encuentra ubicado en otra localización, incluso desconocida, pero que son parte de una red de suministros que comparten los mismos objetivos.
<b>Fallas</b>	Es un problema en un programa de computador o sistema de software que desencadena un resultado indeseado.

<b>Gestión de recursos</b>	Es el proceso de planificar, programar y asignar previamente los recursos para maximizar su eficacia.
<b>GNU</b>	Componentes del Sistema operativo Linux.
<b>Go</b>	Es un lenguaje de programación concurrente y compilado, desarrollado por Google, inspirado en la sintaxis de otros como C, Python o C++.
<b>Guest</b>	Término que se traduce como invitado, el cual se asocia a las máquinas virtuales que se levantan sobre algún anfitrión.
<b>Hipervisor</b>	Es un proceso que crea y ejecuta máquinas virtuales, permite que un ordenador <i>host</i> preste soporte a varias máquinas invitadas mediante el uso compartido de sus recursos.
<b>Host</b>	Término que se traduce como anfitrión y es utilizado para nombrar a los equipos principales que albergan en su interior máquinas virtuales.
<b>Hosting</b>	Servicio en línea que permite publicar un sitio o aplicación web en la red de internet.
<b>IAAS</b>	Infraestructura como servicio ( <i>Infrastructure as a Service</i> ).
<b>Intel</b>	Empresa desarrolladora de circuitos.

<b>ISO</b>	Estandarización de la organización internacional.
<b>Kernel</b>	Es el centro esencial de un sistema operativo de computadora, es el núcleo que proporciona servicios básicos para las demás partes.
<b>Kubernetes</b>	Es una plataforma de orquestación de contenedores de código abierto que automatiza muchos de los procesos manuales involucrados en la implementación, administración y escalado de aplicaciones en contenedores.
<b>KVM</b>	Máquina virtual basada en Kernel ( <i>for Kernel-based</i> ).
<b>Línea de Comandos</b>	Es un tipo de interfaz de usuario de computadora que permite a los usuarios dar instrucciones a algún programa informático o al sistema operativo por medio de una línea de texto simple.
<b>LKVM</b>	Máquina virtual basada en Kernel con Linux Nativo.
<b>LXC</b>	Conjunto de herramientas, plantillas, bibliotecas y enlace de idiomas.
<b>LXD</b>	Es un administrador de contenedores del sistema de próxima generación.



<b>Malware</b>	Es un software o código malicioso, que está diseñado para infiltrarse en un sistema con el fin de dañar o robar datos e información.
<b>Manual</b>	Conjunto de instrucciones, para operar el objeto, software o máquina en cuestión.
<b>Máquina Virtual</b>	Son las instancias de máquinas que no se encuentran presentes, más bien son rentados y utilizados por medio de la red de internet.
<b>Memoria RAM</b>	Memoria de acceso aleatorio, es un tipo de almacenamiento principal de los equipos donde se almacenan programas y datos informativos.
<b>Microsoft Office</b>	Es una familia de software de cliente, de servidor y servicios desarrollados por Microsoft.
<b><i>Migración de Servicios</i></b>	Se le conoce así a la acción de trasladar los servicios o procesos que están detrás de la funcionalidad de algún sistema de un servidor a otro.
<b>Moodle</b>	Es una plataforma de aprendizaje diseñada para proporcionar a educadores, administradores y estudiantes un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados.

<b>Namespace</b>	Un espacio de nombres es un contenedor abstracto en el que un grupo de uno o más identificadores únicos pueden existir.
<b>Nativo</b>	Que tiene una relación desde el momento de la concepción o desarrollo, o que soporta el <i>software</i> o <i>hardware</i> desde su misma implementación.
<b>NIST</b>	Instituto nacional de estándares y tecnología a ( <i>National Institute of Standards and Technology</i> ).
<b>Nube</b>	Es una red mundial de servidores, cada uno con una función única.
<b>On Premise</b>	Se le conoce a las soluciones locales donde se instalan o ejecutan en computadoras que pertenecen a la persona u organización.
<b>Open Source</b>	Es un término que se refiere a que el código de software es de acceso público, por lo tanto cualquiera puede verlo, modificarlo y distribuirlo.
<b>PAAS</b>	Plataforma como servicio ( <i>Platform as a Service</i> ).
<b>PID</b>	Conjunto independiente de ID de procesos.
<b>Procesador</b>	Este se conoce también como CPU, que es una unidad de procesamiento encargada de interpretar

las instrucciones de un hardware haciendo uso de distintas operaciones aritméticas y matemáticas.

**Procesamiento**

Aplicación sistémica de una serie de operaciones sobre un conjunto de datos que la mayoría de ocasiones se realiza por medio de máquinas.

**Proxmox**

Es una plataforma de código abierto completa para la virtualización empresarial.

**Proxmox**

Es una plataforma de virtualización de nivel empresarial 100 % libre y sin límites en su uso.

**QA**

Aseguramiento de calidad (*Quality Assurance*).

**QEMU**

Virtualizador y emulador de máquina genérico y de código abierto.

**Red ubicua**

Es una teoría de computación que se basa en la invisibilidad, la escala local, la ocultación de los desniveles de acondicionamiento y el uso inteligente de los espacios.

**Respaldo**

Permite administrar y controlar distintas copias de archivos que se manejan de forma digital y son importantes, por medio de diversos dispositivos.

**REST**

Estilo Arquitectónico de Software (*Representational state transfer*).

<b>rkt</b>	Motor de contenedores de aplicaciones desarrollado para entornos nativos de la nube de producción moderna.
<b>SAAS</b>	Software como servicio (Software as a Service).
<b>SarsCov2/COVID 19</b>	Es la enfermedad causada por el nuevo coronavirus conocido como SARS-CoV-2. La OMS tuvo noticia por primera vez de la existencia de este nuevo virus el 31 de diciembre de 2019, al ser informada de un grupo de casos de «neumonía vírica» que se habían declarado en Wuhan.
<b>Servidor</b>	Se trata de una o más máquinas capaces de atender las peticiones realizadas por un cliente y devolverle una respuesta en relación a su consulta.
<b>Sistema de Archivos</b>	Es el sistema de almacenamiento de un dispositivo de memoria, que estructura y organiza la escritura, búsqueda, lectura, almacenamiento, edición y eliminación de archivos de una manera concreta.
<b>SOA</b>	Arquitectura orientada a servicios.
<b>systemd</b>	Conjunto de bloques de construcción básicos para un sistema Linux.
<b>Tarea</b>	Es una acción que debe realizarse dentro de un proceso.

<b>Tecnología</b>	Es la aplicación de la ciencia a la resolución de problemas en concreto, por medio del diseño, creación de bienes o servicios que facilitan ciertas tareas.
<b>TI</b>	Tecnología de la información.
<b>Topología de Red</b>	Una topología de red es la disposición de una red, incluyendo sus nodos y líneas de conexión.
<b>UTS</b>	Espacio de nombres Unix ( <i>Unix Timesharing</i> ).
<b>VE</b>	Entorno virtual ( <i>Virtual Environment</i> ).
<b>Versatilidad</b>	Es la cualidad de realizar cosas o acciones distintas, o bien adaptarse fácilmente a diferentes situaciones.
<b>Videoconferencias</b>	Es un sistema interactivo que permite a varios usuarios mantener una conversación virtual por medio de la transmisión en tiempo real de video, sonido y texto a través de internet.
<b>Virtualización</b>	Crea un entorno informático simulado, o virtual, en lugar de un entorno físico.
<b>VMM</b>	Administrador de máquina virtual ( <i>Virtual Machine Manager</i> ).

**YouTube**

Sitio web de origen estadounidense dedicado a compartir videos, presenta una variedad de clips de películas, programas de televisión y videos musicales, así como contenidos amateur como videoblogs y YouTube Gaming.

## RESUMEN

Se utilizó una herramienta de virtualización llamada Proxmox para la administración de máquinas virtuales, en la que cada máquina virtual estará representando una unidad académica, así mismo se hizo uso de Docker para la administración y creación de contenedores para mayor facilidad en cuanto a la implementación de aplicaciones para contar con las configuraciones adecuadas y así crear plantillas que contengan todas las herramientas necesarias para poder copiarlas y levantarlas en cualquier instante que se desee por la División de Educación a Distancia de Entornos Virtuales DEDEV, el fin del uso de esta herramienta propuesta en este documento es lograr facilitar a la DEDEV la migración de un servidor a otro, ya que este proceso se realiza año con año y se buscaba contar con una herramienta que tenga la capacidad de realizar lo que solicitan por lo que se propuso dicha herramienta; ya que como se describe en la arquitectura, el software de virtualización estaría como un sistema operativo, corriendo sobre este, máquinas virtuales, que a su vez tengan contenedores creados corriendo independientemente, manteniendo una administración óptima en el reparto de los recursos y una acertada evaluación en el desempeño de la plantilla.





## OBJETIVOS

### General

Analizar e identificar las distintas posibles soluciones basadas en las herramientas actuales de tecnología que más se adapten a las necesidades y requerimientos que cuenta la División de Educación a Distancia en Entornos Virtuales de la Universidad de San Carlos de Guatemala con el fin de que puedan implementar un *software* para la administración de recursos de los servidores que permitan un óptimo rendimiento.

### Específicos

1. Analizar las herramientas de virtualización de servidores para la administración de los recursos mediante la gestión de una máquina virtual por cada unidad académica en los servidores de la división.
2. Encontrar un método para facilitar la gestión de máquinas virtuales mediante plantillas previamente configuradas.
3. Comparar las distintas propuestas para el uso de una herramienta para el despliegue de aplicaciones dentro de contenedores de *software* para la ejecución de las aplicaciones web que utiliza cada unidad académica de la división que se ajuste a las necesidades de la división de estudios a distancia.

4. Elaborar un documento que contiene el manual para las configuraciones de la herramienta de virtualización de servidores y para las configuraciones de la herramienta de despliegue de aplicaciones.

## HIPÓTESIS

El análisis y estudio de las diferentes herramientas de *software* para la virtualización, permite identificar una solución para mejorar la administración de los recursos del servidor de la División de Educación a Distancia en Entornos Virtuales, DEDEV, por medio de una comparación de propuestas en las cuales sea posible identificar cuál de todas se adapta más a las necesidades y solicitudes presentadas, para que con ello se tenga una mayor factibilidad de implementación futura.



## INTRODUCCIÓN

La computación en la nube o *cloud computing* es una alternativa al uso de servidores locales de la empresa. Este servicio surge del excedente de equipos y capacidad de cómputo que tienen algunas empresas, permitiéndoles alquilar aquellos no utilizados y convertirlos en un ingreso. Para alquilar estos recursos se necesita de la virtualización, que permite crear diferentes servicios y distribuirlos entre diferentes usuarios, aprovechando la capacidad de cómputo de un servidor de la mejor forma posible.

En la actualidad, organizaciones, tanto privadas como públicas, tienen inconvenientes para desplegar sus herramientas de software en la nube, teniendo que optar por un modelo tradicional en servidores o equipos de cómputo locales, también conocido con el término de modelo *on premise*. Algunos de estos inconvenientes son: regulaciones gubernamentales, capacidad económica, falta de conocimiento o experiencia en soluciones alternativas, por mencionar algunos de los más comunes.

La investigación pretende proponer una solución a la División de Educación a Distancia en Entornos Virtuales, DEDEV, para virtualizar los recursos de cómputo con los que cuentan actualmente, con herramientas de software que no incurran en gastos económicos, que faciliten la administración de estos, incluyendo su migración cuando deseen cambiar de proveedor de servicios de computación en la nube.

Este esquema funcionaria para colocar dentro del servidor, un gestor de contenidos académicos, la cual administraría y sería utilizada por las distintas

unidades que conforman la DEDEV, por lo tanto, se deben de repartir los recursos de forma equitativa para que cada servicio o sitio que se ejecute para los estudiantes involucrados funcione sin ningún inconveniente.

# 1. FUNDAMENTOS TEÓRICOS

Es de vital importancia conocer los conceptos que se desarrollarán en los capítulos siguientes con el fin de comprender a totalidad la solución que se plasma en un manual como resultado del trabajo en conjunto que realizó el equipo de desarrollo de este documento, por lo cual el capítulo está dedicado a describir el concepto detallado y concreto de cada herramienta que ha formado parte de la concepción de la solución propuesta, con lo cual se desea transmitir una claridad en la ejecución que de ser implementado sobre los estándares que se manejan durante el desarrollo de este documento, permita comprender la idea central del mismo:

## 1.1. Virtualización

La virtualización es una tecnología que se utiliza para crear servicios de TI simulando recursos de *hardware*, a través de *software*, como sistemas operativos, dispositivos de almacenamiento, dispositivos de redes, entre otros.

Al distribuir los recursos a los diferentes sistemas o dispositivos simulados, permite optimizar el uso de la mayor parte de los recursos de *hardware* por cada uno de los sistemas o dispositivos simulados.

El *software* encargado de simular y distribuir los recursos de *hardware* se conoce como hipervisor. Los hipervisores pueden estar instalados sobre un sistema operativo o es posible instalarlos directamente sobre el *hardware*.

### **1.1.1. Uso**

Algunos de los usos más comunes de la virtualización son:

- Consolidación de equipos virtuales en un solo equipo físico.
- Mitigar fallas mediante el respaldo de equipos físicos a través de equipos virtuales.
- Uso de aplicaciones heredadas que requieran *hardware* desactualizado para poder ser ejecutadas.
- Creación de ambientes de pruebas que no comprometan el funcionamiento o la seguridad de los ambientes en ejecución.

### **1.1.2. Tipos**

En la actualidad, existe una gran variedad de tipos de virtualización en el mercado, pero se tomaron en consideración los que son de mayor relevancia o tienen una mayor aplicación, los cuales se detallan en los siguientes puntos:

#### **1.1.2.1. Virtualización de servidores**

Permite ejecutar una cantidad de sistemas operativos, para realizar diferentes tareas asignadas a cada servidor virtualizado, sobre un solo servidor físico. Esto, para aumentar una mayor eficiencia de procesamiento, ejecutar funciones específicas en cada servidor virtual.

#### **1.1.2.2. Virtualización de escritorios**

La virtualización de escritorios permite centralizar las aplicaciones o escritorios virtualizados, mediante una herramienta de administración. La



administración central no está limitada a una sola máquina física, sino a varias máquinas físicas de forma simultánea.

La virtualización de escritorios permite instalar, configurar, actualizar, las diferentes aplicaciones o escritorios virtuales de forma masiva, a cada uno de ellos.

#### **1.1.2.3. Virtualización de sistemas operativos**

Este tipo de virtualización se realiza sobre el kernel, o núcleo, permitiendo ejecutar diferentes sistemas operativos de forma paralela.

#### **1.1.2.4. Virtualización de las funciones de red**

Permite separar las funciones importantes para poderlas distribuir en los diferentes entornos, reduciendo la cantidad de equipos físicos necesarios para tipologías de redes independientes.

#### **1.1.3. Ventajas**

- Mejora de la utilización de recursos físicos de *hardware*.
- Ahorro de costos al no tener que comprar más equipo.
- Ahorro de costos al hacer mejor uso del equipo.
- El aislamiento y la seguridad que esto conlleva.
- Mejora del consumo energético al no ser una solución *on premise*.

#### **1.1.4. Desventajas**

- Un menor rendimiento de las aplicaciones o *hardware* virtualizado.

- Limitación del servicio o *hardware* a virtualizar, debido a la compatibilidad con el hipervisor.
- Colapso de los servicios o *hardware* virtualizados que se alojan en un único servidor.
- Dificultad de migración de servicios o *hardware* virtual al no existir una norma para la portabilidad o migración, debido a la gran cantidad de hipervisores existentes.

## **1.2. Hipervisor**

También conocido como VMM, por sus siglas en inglés Virtual Machine Manager, y su traducción al español es Administrador de Máquina Virtual. Es una herramienta de *software* encargada de la administración de máquinas virtuales, desde su creación, ejecución y la administración de recursos de estas.

La tarea de administración de recursos es vital dado que el Hipervisor se encarga de gestionar los recursos, como la CPU, dispositivos de memoria y almacenamiento, por mencionar algunos; del equipo anfitrión o “host” y distribuirlos a los diferentes equipos virtualizados o alojados o “guest”.

### **1.2.1. Tipos de hipervisores**

En la actualidad, existe una gran variedad de tipos de hipervisores en el mercado, pero se tomaron en consideración los que son de mayor relevancia o tienen una mayor aplicación, los cuales se detallan en los siguientes puntos:

### **1.2.1.1. Hipervisor tipo 1**

También conocido como Hipervisor nativo o "unhosted" o sin sistema operativo. Recibe este nombre dado que el hipervisor se ejecuta directamente en el equipo físico sin necesidad de una capa intermedia entre el *hardware*.

### **1.2.1.2. Hipervisor tipo 2**

Recibe el nombre de Hipervisor alojado, que a diferencia de su contraparte, el Hipervisor Tipo 1, el Hipervisor Tipo 2 necesita un sistema operativo como la capa intermedia para el acceso a *hardware*.

### **1.2.1.3. Hipervisor Híbrido**

En este tipo de hipervisor, ambos, el sistema operativo anfitrión y el hipervisor interactúan sobre el hardware de forma directa. Así mismo, las máquinas virtuales se ejecutan en un nivel superior.

## **1.2.2. Ventajas**

- La administración de múltiples máquinas virtuales se realiza de manera centralizada, mediante un único panel de administración.
- Al centralizar la administración de recursos se evitan los conflictos por el uso compartido de los mismos, entre las diferentes máquinas virtuales existentes.
- Permite ejecutar diferentes tipos de sistemas operativos según lo requerido para cada máquina virtual.

- Los errores ocurridos en cada máquina virtual existente no afectan al resto de máquinas virtuales existentes.
- Al no estar ligadas a un *hardware*, las máquinas virtuales alojadas en el hipervisor se pueden migrar con mayor facilidad, entre hipervisores compatibles.

### **1.2.3. Desventajas**

- Si el hipervisor se ve afectado por una falla, todas las máquinas virtuales se verán afectadas también.
- Las máquinas virtuales administradas por un hipervisor tienen menor rendimiento, dado que dependen de una capa intermedia de acceso a los recursos.
- La migración de máquinas virtuales entre diferentes hipervisores no es compatible.

## **1.3. Máquina virtual**

Hablar de máquina virtual es hablar de un dispositivo que contiene las mismas características que una computadora real, la diferencia radica en que una máquina virtual no está expuesta físicamente como una computadora de escritorio o personal.

En sí, una máquina virtual es un *software* que simula todas las características y cualidades de una computadora, la cual pueden imitar una arquitectura, hasta ejecutar programas y aplicaciones como una computadora

real. Las máquinas virtuales también poseen memoria RAM, disco duro, entre otras características similares a las de una real.

### **1.3.1. Uso**

Una de las principales funciones de las máquinas virtuales es alojar sistemas operativos distintos al que se tiene en la máquina *host*, ya que se utiliza de esta forma para poder probar nuevas funcionalidades y realizar las configuraciones que se deseen sin afectar al sistema principal.

Las máquinas virtuales también se utilizan para realizar tareas y funciones específicas sin poner en riesgo la máquina física o *host* en la que se encuentra alojada, porque muchas veces lo que se busca es hacer experimentos en el sistema, pero los cambios que se hagan en la máquina virtual no puede afectar al entorno *host*, porque lo que hace es compartir los mismos recursos, sin embargo, no los puede afectar.

### **1.3.2. Tipos de máquinas virtuales**

En la actualidad, existe una gran variedad de tipos de máquinas virtuales en el mercado, pero se tomaron en consideración los que son de mayor relevancia o tienen una mayor aplicación, los cuales se detallan en los siguientes puntos.

#### **1.3.2.1. Máquinas virtuales de procesos**

Las máquinas virtuales de procesos también son conocidas como máquinas virtuales de aplicación, esta se ejecuta como un proceso normal dentro de un sistema operativo, el cual sirve como puente para que se pueda

interpretar entre código fuente y código máquina, permitiendo así que un programa se ejecute siempre de la misma forma sobre cualquier plataforma.

### **1.3.2.2. Máquinas virtuales de sistema**

La principal función de estas máquinas virtuales es que de un mismo ordenador *host* pueden compartir los recursos a varias máquinas virtuales, lo cual hace que sean poco eficientes ya que si se tienen varias máquinas corriendo bajo un mismo ordenador *host* compartiendo el mismo recurso esta función las hace más lentas y con muy pocos recursos entre ellas.

### **1.3.3. Ventajas**

- Proporcionan opciones de recuperación ante desastres.
- Se puede instalar cualquier sistema operativo y realizar las configuraciones que se deseen sin afectar el sistema principal.
- En muchas ocasiones ayudan a protegernos del malware.
- Facilita el traslado de un sistema operativo de un ordenador a otro.
- Es una tecnología segura y sencilla de utilizar.
- Ofrecen puntos de restauración.

### **1.3.4. Desventajas**

- Una de las limitantes es el recurso proporcionado por la máquina *host* haciendo que el rendimiento de esta se vea afectado por lo que esto hará que no funcione como la máquina física.

- La ejecución de varias máquinas virtuales también es una limitante, porque éstas comparten recursos por lo que las hace menos eficientes y con mayor tiempo de respuesta en algunas ocasiones.

#### **1.4. Gestor de contenidos**

Es un programa o incluso aplicación en algunos casos web, que permite la creación de cierta estructura para el soporte de la creación, administración y publicación de contenido o bien de servicios, para que su uso sea más ordenado y el mantenimiento del sistema completo sea modular, de tal forma que sea manejable y escalable, incluso.

Este tipo de programas permite al usuario tener un control directo y más organizado de los servicios que va a proveer su servidor, en el caso de su contenido, ser máquinas virtuales, le da la oportunidad de manejar un sistema que administre de forma conjunta todas las que se encuentren dentro de este entorno, así como contenedores y otros tipos de segmentación que pudieran utilizarse para la provisión de utilidades.

##### **1.4.1. Uso**

Es utilizado para el manejo de tareas asociadas con la administración de aplicaciones contenerizadas o bien por máquinas virtuales, así como componentes de aplicación. Las aplicaciones se envasan en estos tipos de componentes para facilitar el escalado, duplicación e incluso la actualización. Este *software* sirve como gestión del contenido, proveyendo así a la empresa un gran número de instancias, evitando en una gran parte el trabajo básico y garantizando así la interoperabilidad de los sistemas.

### **1.4.2. Ventajas**

- Se tiene un grado de escalabilidad mayor: este es un factor determinante, mayormente cuando se renta un espacio en la nube, puesto que permite tener un mejor dominio de los recursos.
- Ahorro de costos: con estos *softwares* de gestión se logra aprovechar al máximo los recursos, y en dado caso, ya una máquina, contenedor o componente que ya no se está usando, se puede liberar para otras tareas o dejarlo inactivo.
- Se obtiene un sistema autónomo y veloz, debido a que la escalabilidad del *software*, en dado caso que se aumente la demanda, se tiene el control para proveer de más recursos y mantener siempre un sistema funcional.
- Permite a los administradores del sistema tener una visibilidad completa y mayor seguimiento puesto que le brinda un control absoluto, fácil e intuitivo con el cual es factible tener el desarrollo completo del soporte en la aplicación.

### **1.4.3. Desventajas**

- Se tiene un sistema que puede ser vulnerable puesto que de tener acceso al servidor donde se tiene el gestor de componentes tiene acceso a todas las instancias de la solución.
- Se tiene un sistema modular que requiere de conocimiento técnico para existir comunicación entre las distintas instancias que se crean.



- Se tiene el control completo de los componentes lo cual deja una brecha para que puedan existir errores y eso provoque desestabilidad.
- Se necesita tener conocimiento del *software* y el manejo de la herramienta que se seleccione.

## **1.5. Contenedores**

Es la tecnología que permite al usuario empaquetar y tener de forma aislada las aplicaciones con su respectivo entorno de tiempo de ejecución, en otras palabras, este paquete contendrá todos los archivos o herramientas que requiera la aplicación para ejecutarse sin agregar ningún otro consumo de recursos adicional, lo que permite movilizar la aplicación que se encuentra dentro de este sin el riesgo que se pierda alguna funcionalidad.

### **1.5.1. Uso**

Los contenedores se utilizan para dar una garantía sobre las aplicaciones que se ejecutan dentro de los mismos, puedan correr de forma correcta cuando se cambia el entorno sobre el que trabajan, o que no se da algún fallo en su estructura o de algún otro tipo, tienen una semejanza en ello con la virtualización, con la diferencia que en estos casos no se requiere la instalación y el uso de recursos para levantar un sistema operativo, sino la imagen utilizada debería contener únicamente las herramientas que se requieren para el buen funcionamiento de la aplicación.

Uno de los factores que influye en el uso de los contenedores radica en la utilidad y agilidad al momento de realizar alguna migración, o bien la construcción de un ambiente de desarrollo o QA a partir de otro, puesto que en

estos casos, es posible que el código no funcione de la misma forma a donde fue implementado inicialmente, caso contrario si se ingresa en un contenedor puesto que este puede portarse conforme más convenga.

## **1.5.2. Tipos**

Se tienen definidos 3 tipos como tal que se logran identificar, adicional existen desarrollos paralelos con los cuales funcionan como parte de los descritos a continuación:

### **1.5.2.1. rkt**

Se considera uno de los mayores competidores en el área del mercado de la virtualización basado en contenedores de ejecución, este funciona con un núcleo de Linux necesario, el cual puede ser cualquiera que cumpla con ser amd64 moderno.

Este tipo de contenedores son compatibles con distintas distribuciones, tales como: Arch Linux, CentOS, CoreOS, Debian, Fedora, NixOS, openSUSE, Ubuntu y Void, así como puede incluso funcionar en macOS, inclusive en Windows con alguna herramienta de virtualización como Vagrant Virtual Machine.

Este está desarrollado en Go y tiene una licencia de Apache 2.0, no requiere de algún tipo de operación secundaria, sino trabaja con sistemas init consolidados como systemd y upstart, los cuales utiliza para iniciar y administrar contenedores, adicional a ello rkt, no depende únicamente de funciones del *kernel* de Linux como lo son Cgroups y Namespaces para la virtualización de aplicaciones. Con el entorno de ejecución de CoreOS también

se pueden iniciar contenedores como máquinas virtuales sobre la base de KVM (LKVM o QEMU) o de la tecnología Clear Container de Intel.

### **1.5.2.2. LXC**

También conocidos como Linux Containers, es otra alternativa de contenedores, donde se maneja como un juego de herramientas, plantillas, bibliotecas y bindings de lenguaje que, van a constituir una interface de espacio de usuario para las funciones nativas de virtualización con contenedores que tienen un núcleo Linux, adicional ofrece a sus usuarios una manera sencilla e intuitiva de crear y administrar los contenedores con los diversos software y sistemas.

Para aislar los procesos entre sí, LXC utiliza los siguientes métodos de aislamiento:

- Espacios de nombres (namespaces) IPC, UTS, Mount, PID, de red y de usuario.
- Cgroups
- Perfiles AppArmor y SELinux
- Reglas Seccomp
- Chroots
- Kernel capabilities

Su principal objetivo es desarrollar un espacio para los contenedores que se diferencie lo más mínimo posible de una instalación normal de Linux.

### **1.5.2.3. LXD**

Linux llevó a la luz la alternativa de contenedores LXD, el cual es un sucesor de LXC, toma gran parte del núcleo de contenedores de Linux pero amplía el demonio LXD, por lo que se le llega a considerar hipervisor de contenedores. Se compone de tres elementos: el demonio LXD, la terminal de cliente LXC y un plugin nova-compute-lxd, donde la comunicación entre cliente y demonio se realiza por medio de REST-API.

Trabaja igual que su predecesor LXC en cuanto al núcleo de Linux, estando disponible para las distribuciones:

- Para el cliente de líneas de comando: Ubuntu 14.04 LTS y Ubuntu 16.06 LTS.
- Para nova-compute-lxc Ubuntu 16.04 LTS.

No es compatible con otras plataformas, teniendo un formato idéntico a LXC, con licencia Apache 2.0 y desarrollado en Go.

### **1.5.2.4. Docker**

Se le reconoce como el líder tecnológico en el mercado de virtualización basado en contenedores, debido a la resonancia de los medios y su ecosistema en constante crecimiento, este recurre a las características fundamentales del núcleo de Linux para el aislamiento de procesos entre si y debido a los entornos de ejecución que posee, son lo que le permiten que funcione en paralelo en contenedores aislados sin necesidad de gastar algún recurso de la máquina, se consolida como la mejor alternativa de virtualización basado en hipervisores.

Docker requiere un núcleo de Linux en una versión 3.10 o superior, el mismo soporta distribuciones de Linux con su versión Community Edition: Ubuntu, Debian, CentOS, Fedora, por su parte con Docker Enterprise Edition EE: se tiene Ubuntu, Red Hat Enterprise Linux, CentOS, Oracle Linux y SUSE Linux Enterprise Server, adicional a ello se puede identificar que otras plataformas que maneja son: Docker Community Edition (CE): Microsoft Windows 10 (Pro, Enterprise o Education con 64 Bit), macOS (Yosemite 10.10.3 o superior), Microsoft Azure, Amazon Web Services (AWS), Docker Enterprise Edition (EE): Microsoft Windows Server 2016, Microsoft Windows 10 (Pro, Enterprise o Education con 64 Bit), Microsoft Azure, Amazon Web Services (AWS).

### **1.5.3. Ventajas**

- **Modularidad:** el enfoque que se maneja para la creación de contenedores se basa en la capacidad de tomar parte de una aplicación, actualizarla o repararla sin tomarla completamente, al ser un enfoque de microservicios, se pueden compartir procesos entre diferentes aplicaciones de la misma forma en que funciona SOA.
- **Capas y control de versión de la imagen:** cada imagen se conforma por una serie de capas que se combinan en una única solución de imagen, cada capa es creada al cambiarse dicha imagen, estas capas son reutilizables para nuevos contenedores lo cual hace mucho más efectivo su proceso de construcción.
- **Restauración:** se puede manejar las versiones, a partir del manejo de las capas, esto soporta un enfoque de desarrollo ágil y ayuda a hacer la

integración e implementación continua desde una perspectiva de herramientas.

- Implementación rápida: se pueden reducir los tiempos de implementación a segundos, al crear un contenedor para cada proceso puede compartir rápidamente los procesos similares con nuevas aplicaciones.

#### **1.5.4. Desventajas**

- Cuando se comienza a usar cada vez más contenedores y aplicaciones en estos es más difícil hacer la separación en cientos de piezas, la administración y la orquestación. Finalmente se tendrán que agrupar contenedores para ofrecer servicios. Para ello funcionan servicios de gestores de contenido como Proxmox o bien Kubernetes.
- No se tiene la capacidad de usar procesos como `ron` o `syslog` dentro de un contenedor, junto con la aplicación, existen límites como la limpieza de procesos después de terminar con los procesos hijo.

### **1.6. Cloud Computing**

La computadora en la nube puede definirse más como una forma de negocio, el cual ofrece servicios a través de internet, en donde su principal objetivo es que los clientes no tengan esa preocupación de contar con un espacio físico con la capacidad necesaria o suficiente para poder almacenar toda la información que posea.

El Instituto Nacional de Estándares y Tecnología (*National Institute of Standards and Technology* - NIST) define a la computación en la nube como un

modelo para permitir el acceso de red ubicuo, conveniente y bajo demanda a un grupo compartido de recursos informáticos configurables, como por ejemplo: redes, servidores, almacenamiento, aplicaciones y servicios. Estos recursos configurables se pueden aprovisionar y liberar rápidamente con una administración mínima de esfuerzo o interacción del proveedor de servicios.

También puede definirse por ser una tecnología que busca tener todo centralizado en una red amplia en la que se encuentra información, procesos, aplicaciones y lo más importante es que permite optimizar y ahorrar mucho espacio en el disco duro.

Hablar de *cloud computing* es hablar de cambios, de evolución, ya que en la actualidad se busca tener servicios en la nube desde una página web hasta un sitio completo para poder hacer uso de este y sin tener que tenerlo en un ordenador físico, y todo esto gracias a que la tecnología avanza de forma escalable. También ofrece un amplio escenario de recursos para poder utilizarlos a la necesidad del cliente o empresa, ya que cuenta con fácil acceso, mantenimiento, seguridad y flexibilidad en la información que es lo que hoy en día se busca, que los clientes puedan ingresar en cualquier hora y en cualquier lugar a revisar sus datos.

### **1.6.1. Uso**

La tecnología en la nube ofrece demasiados servicios para su uso, desde enviar un correo electrónico, almacenar imágenes, escuchar música en una plataforma en específico, todo esto es gracias a que existe *cloud computing*. Entre los servicios más comunes que posee la tecnología en la nube podemos encontrar:

- Almacenar información
- Crear aplicaciones a petición del cliente
- Creación de sitios web
- Subir videos a una plataforma en específico
- Comunicación a distancia (videoconferencias)
- Educación en línea
- Crear servidores con los requerimientos del cliente o empresa
- Arquitecturas complejas

Y así se puede listar muchos usos para la computación en la nube, todo depende de qué servicios desee solicitar la organización a su proveedor, ya que se puede solicitar desde un *hosting* que no es más que un almacenamiento web para las personas que se dedican a realizar páginas web o a escribir un blog, es esencial que se conozca un poco sobre este tema ya que existen desde *hosting* compartidos hasta *hosting* privados los cuales destacan entre los usos más comunes de la computación en la nube, ya que si una organización desea crear un servidor con cierto sistema operativo y con ciertas capacidades y recursos, todo esto es posible gracias a que existe un tipo de *hosting* llamado servidor dedicado, el cual no comparte los recursos del servidor con otras personas u organizaciones y esto lo hace mucho más efectivo en cuanto tráfico que puede causar al ser consumido algún servicio web por muchos clientes, también cuenta con mayor seguridad.

### **1.6.2. Tipos o modelos de servicio en la nube**

Estos modelos son diseñados según la necesidad del cliente y para ello se cuenta con los siguientes modelos de servicio que la nube ofrece para una mejor implementación.



### 1.6.2.1. Software como servicio (SaaS)

Este método es el encargado de proporcionar el acceso a *software* y sus funciones de forma remota como un servicio basado en la web pagando únicamente por los recursos que son utilizados por lo que lo convierte en una gran ventaja ya que se puede escalar o reducir verticalmente los recursos en función del uso.

En este escenario es donde se presenta la disponibilidad de acceso, ya que se puede acceder desde cualquier parte con cualquier dispositivo que cuente con conexión a internet.

Por lo que se puede definir que SaaS es un proceso de entrega de *software* que permite acceso a aplicaciones remotas como un servicio basado en la web, sin necesidad de instalar algún tipo de *software*, basta con ejecutarlas directamente desde un explorador web.

Figura 1. Disponibilidad de acceso



Fuente: elaboración propia, empleando draw.io 2000.

### **1.6.2.2. Plataforma como servicio (PaaS)**

Es una plataforma informática que se entrega como un servicio. Estas plataformas como servicio están diseñadas para desarrolladores, el cual tienen como objetivo simplificar el proceso de creación e implementación de *software*, así mismo se puede decir que este servicio es una forma o proceso de alquilar tanto *hardware* como sistemas operativos, la cual permite al cliente utilizar servicios en los que se puedan ejecutar aplicaciones existentes o servicios que él mismo desarrolle.

Algunas de las virtudes de la plataforma como servicio es que incluyen base de datos, interfaz de desarrollo, ambientes de pruebas y otros recursos de computación alojando todo en un mismo entorno.

Otra de las características es que el proveedor ofrece su propia infraestructura evitando así que los usuarios u organizaciones se centren únicamente en el desarrollo, así mismo estos proporcionan lo que es la seguridad, el sistema operativo y por si fuera poco las copias de seguridad.

Figura 2. **Plataforma de servicio**



Fuente: elaboración propia, empleando draw.io 2000.

### 1.6.2.3. **Infraestructura como servicio (IaaS)**

La infraestructura como servicio ofrece recursos informáticos a los usuarios en el que se pueden encontrar desde servidores individuales hasta servidores compartidos, así mismo almacenamiento, servidores de correos, servidores de nombre de dominio, entre otros. En sí el usuario cuenta con muchos recursos a su disposición, pero se puede encontrar con una limitante y es que hay proveedores que ofrecen sus servicios pero con tarifas mensuales o anuales que incluyen la licencia de los servicios que están prestando.

Algunas de las características de esta infraestructura es que puede llegar a ser escalable según el uso o las necesidades de procesamiento y/o almacenamiento, también evita el costo a las organizaciones de comprar y mantener su propio *hardware*.

Figura 3. **Infraestructura de servicio**



Fuente: elaboración propia, empleando draw.io 2000.

### 1.6.3. **Ventajas**

- Ahorro de costos en equipos y sistemas tecnológicos.
- Almacena datos de forma ilimitada y segura.
- Libera el espacio del disco duro de un ordenador físico.
- La flexibilidad que ofrece a las grandes empresas en la forma de acceder a los datos en tiempo y lugar.
- Es multiplataforma.
- Posee la capacidad de que los servicios sean portables.
- Todo lo que se desee hacer es en tiempo real.
- Actualización automática.

#### **1.6.4. Desventajas**

- Acceso a internet: esta es una limitante que se tiene al querer utilizar un servicio en la nube ya que sin acceso a internet no se puede acceder a la información que se tiene alojada en la nube.
- La privacidad juega un papel muy importante en la nube ya que no se sabe con exactitud quién puede tener acceso a dicha información que se pueda almacenar en sus servidores.
- Migración de los datos: este punto es importante tomarlo en cuenta ya que una vez se escoja un proveedor de servicios puede llegar a ser muy complicado el trasladar los datos de una aplicación a otro proveedor de servicios.

#### **1.7. Educación en línea**

Es un tipo de educación donde los participantes, docentes y estudiantes, hacen uso de dispositivos que permiten el acceso a diferentes tecnologías digitales de comunicación a través de internet. La educación en línea promueve un entorno con mayor participación entre los asistentes y otras técnicas para facilitar el aprendizaje.

##### **1.7.1. Uso**

El uso de la educación en línea en la actualidad se utiliza para disminuir la interacción entre personas por la pandemia provocada por el Covid-19. Al estar cerrados algunos espacios públicos, o permitir el ingreso limitado de personas, la educación en línea es una herramienta con mayor auge y utilidad.

La educación en línea no se limita al sector educativo, sino a otros sectores, como el laboral. Los empresarios hacen uso de la educación en línea para la capacitación del personal.

### **1.7.2. Tipos**

Los tipos de educación en línea se pueden definir basándose en cómo es la interacción entre los participantes, docentes y estudiantes.

- Los estudiantes pueden recibir las clases en salones o desde sus hogares. Los profesores están en el salón para atender a los estudiantes.
- Los estudiantes pueden recibir clases dentro de la institución o en sus hogares. El profesor atiende a los estudiantes de forma virtual.
- Los estudiantes y profesores combinan parte de la enseñanza presencial y de forma virtual.
- Los estudiantes reciben todas las clases únicamente desde sus hogares. Los profesores atienden a los estudiantes de forma virtual.

### **1.7.3. Ventajas**

- Flexibilidad geográfica: los estudiantes pueden estar en diferentes ubicaciones donde tengan acceso a un dispositivo que permita conectarse a Internet, disminuyendo total o parcialmente los tiempos por transporte de los estudiantes a los centros educativos, permitiendo el acceso desde cualquier parte del mundo.
- Flexibilidad de horarios: los estudiantes tienen acceso al material didáctico a cualquier hora del día, pero ciertas actividades como atención

y resolución de dudas con el profesor, evaluaciones, por mencionar algunas, requieren un horario establecido.

- La educación en línea se puede implementar en diferentes ámbitos, por ejemplo, en el ámbito laboral para capacitación de empleados.

#### **1.7.4. Desventajas**

- Requiere de un esfuerzo mayor dado que no todos los estudiantes mantienen la atención, consultan el material, realizan las actividades en esta modalidad.
- Disminuye la interrelación entre los alumnos y profesores.
- Algunos estudiantes que no tienen el mismo nivel de acceso a los dispositivos de comunicación, internet, por ejemplo, que los demás. Esto afecta la calidad educativa.
- En algunos casos puede suponer un coste económico elevado o fuera del alcance de los estudiantes.
- La dificultad que conlleva realizar actividades supervisadas como las evaluaciones, trabajos en grupos, por ejemplo.





## **2. ANÁLISIS Y COMPARACIÓN DE SOLUCIONES**

Actualmente, en la División de Educación a Distancia en Entornos Virtuales de la Universidad de San Carlos de Guatemala, se adquiere un espacio en la nube con un proveedor de servicios en específico para levantar un servidor e instalar una plataforma web de acceso público. El contrato de estos servicios tiene una duración de un año, por lo que el proceso de migración de la plataforma web y los datos es tedioso y repetitivo debido al cambio de entorno o nodo prestado por el proveedor, por la gran cantidad de información que se maneja hoy en día en la división, y debido a que no cuentan con ninguna herramienta que los ayude a realizar este proceso, se complica el traslado de la información puesto que se requiere levantar desde cero todos los sitios en el nuevo ambiente designado.

Adicional a ello, los días que se realizan actividades que provocan una alta demanda del uso del servidor, como sería, cuando se realizan exámenes en las distintas unidades académicas, se afecta el rendimiento del resto de portales que pertenecen a otras unidades; los demás días, el uso del servidor es mínimo. La División requiere que la solución le permita administrar los recursos del servidor cuando exista una alta demanda, como en el caso explicado con anterioridad.

Debido a lo anterior, es muy importante hacer uso de una herramienta que ayude a realizar estas acciones sin tener que empezar desde cero con las configuraciones de todos los entornos de las distintas unidades académicas que se manejan hoy en día en la división y para ello existen muchas herramientas o formas que pueden ayudar con dicha problemática, así mismo es importante

mencionar que cada una tiene un enfoque diferente al alcance de lo solicitado por la División de Educación a Distancia en Entornos Virtuales.

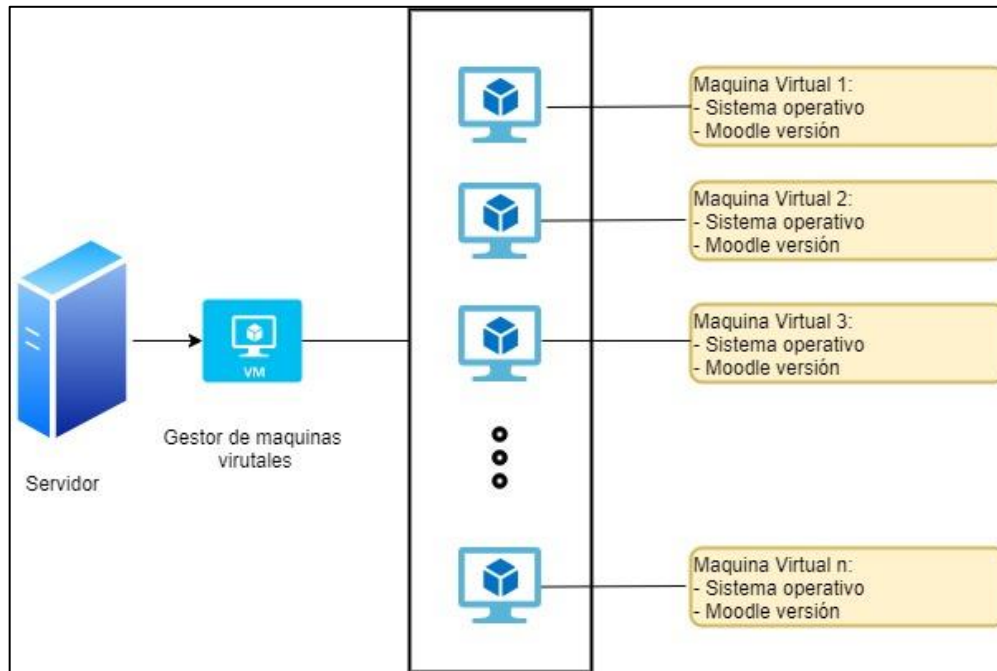
Es importante destacar que la solución propuesta y las alternativas a mencionar están orientadas a los recursos y posibilidades con los que cuenta la División para poder realizar estas modificaciones, por lo tanto debido a esto, se encontrará a lo largo de las distintas opciones solo aquellas herramientas que son gratuitas, puesto que el insumo de licencias no podría contemplarse como parte de los requisitos dentro de la solución.

## **2.1. Máquinas virtuales por cada unidad académica en el servidor (máquinas virtuales como instancias)**

Se pretende utilizar una máquina virtual por cada unidad académica, ya que de esta forma se pueden levantar las instancias que en esos momentos se deseen utilizar, esto con el fin de ahorrar recursos y que se disponga tener listas las máquinas virtuales cuando se requieran.

El proceso con el cual se llevaría a cabo, sería: sobre el servidor entregado por el proveedor se instala el gestor de máquinas virtuales, donde se crea una por cada unidad académica, proveyendo a cada una los recursos de forma equitativa, según la capacidad máxima del servidor central, dejándole a este un máximo de 2 GB de memoria RAM y 8 GB de Disco Duro, siendo el resto distribuido entre las distintas instancias, las cuales estarían funcionando con un Sistema Operativo en el cual se les debería instalar moodle, así como la base de datos y demás sitios pertenecientes a la división que hacen posible la interacción con el estudiante por medio de los sitios web con que se cuenta para su educación.

Figura 4. **Estructura**



Fuente: elaboración propia, empleando draw.io 2000.

### 2.1.1. **Puntos a favor de la solución**

Uno de los principales puntos a favor de esta solución es que se puede tener un fácil manejo de la información debido a que la instancia no está exenta de correr/recibir algún daño que incluso tenga que eliminarse el disco por la información no corre ningún riesgo ya que puede ser trasladada a otra máquina virtual sin ningún problema, puesto que esa es una de las ventajas de utilizar máquinas virtuales “se puede realizar copias de seguridad programadas”.

La solución también brinda de un espacio fijo reservado para cada instancia, lo cual resuelve la problemática que existe en los picos de alta demanda en alguna de las unidades que afecta actualmente a las demás, y

satura todo el nodo, haciendo con esto que en todos los sitios asociados a este servidor se experimenten problemas de lentitud o pérdida temporal del sistema, adicional a ello, se pretende realizar con la ayuda de un gestor de componentes, el cual en este caso, teniendo el rol de modulador, permitiría el control y dominio del administrador de forma general a cada unidad.

### **2.1.2. Puntos en contra de la solución**

La propuesta permite tener ciertas ventajas con respecto al modelo actual, pero de igual forma no cumple completamente los requisitos que se buscan como solución, debido a que en este caso se estaría desperdiciando el espacio que ocupa el sistema operativo de cada máquina virtual, en el cual los recursos estarán estáticos, razón por la cual tiene una gran limitante puesto que no podría haber un balance en las unidades conforme sus picos de demanda, adicional a ello, aunque se cuenta con una administración desde el servidor raíz, el esquema no permite una fácil utilización y control de la solución y para gestionar algún recurso o unidad se convierte en una tarea dificultosa/tediosa/farragosa/engorrosa.

Por último, debido a que se quiere automatizar el traslado de información y las migraciones constantes que se hacen año con año, se logró visualizar que este esquema no proveía de los recursos para facilitar este requerimiento.

## **2.2. Instancias de contenedores necesarias por cada unidad académica en el servidor (contenedores como instancias)**

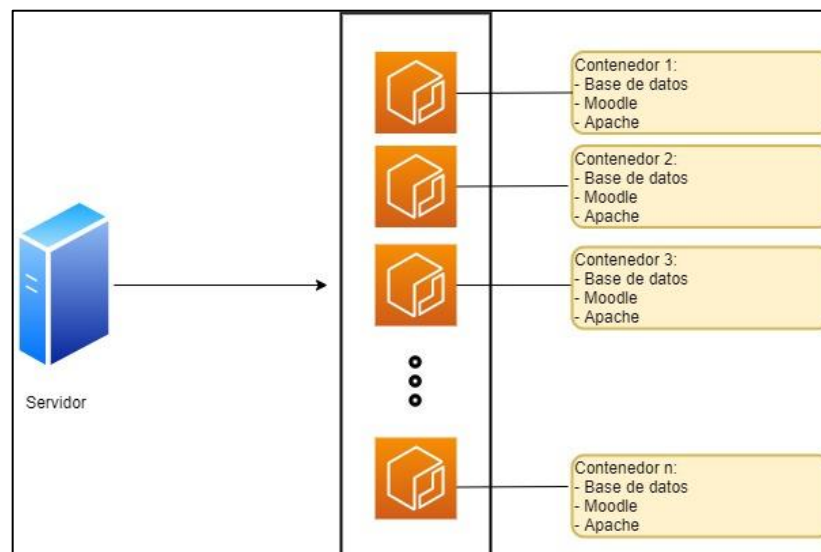
La segunda alternativa propuesta surge a partir de las ventajas que proporcionaba la primera, pero cambiando el esquema de máquinas virtuales por contenedores, donde se tendría uno por cada unidad académica, tal como se planteaba la propuesta anterior. Esta solución está dada a partir de la

premisa de la misma definición de los contenedores, donde se tiene por objeto minimizar el impacto a los recursos consumidos por la misma estructura y proveer de un mayor aprovechamiento.

Este caso se compone de la siguiente forma:

Sobre el servidor que obtiene la división se instala la herramienta para la creación y manejo de contenedores, donde se crearán estos con su respectivo mapeo de puertos sobre el que estarán funcionando los servicios necesarios (base de datos, moodle, apache, entre otros), adicional se realizará un mapeo de directorios para poder manejar y tener un control de la información que se maneje dentro de cada instancia desde el servidor raíz, luego se levantan todos los contenedores y estos estarían aumentando o disminuyendo sus recursos conforme su necesidad y consumo.

Figura 5. **Servidor**



Fuente: elaboración propia, empleando draw.io 2000.

### **2.2.1. Puntos a favor de la solución**

El aprovechamiento de recursos es el principal factor que se puede mencionar con respecto a esta solución, debido a que se toma la idea del encapsulamiento por unidad académica, que se mencionaba en el punto anterior, pero con la mejora del aprovechamiento correcto de cada instancia, puesto que en este caso no se requiere de tener un sistema operativo.

Otro punto que es muy importante de la solución es el manejo automático que tienen los contenedores de los recursos, puesto que se van administrando por ellos mismos según requiera la demanda, lo cual hace que la intervención de un administrador sea mínima y se automatice el proceso.

Para mejorar el trabajo que implican las migraciones, estos contenedores pueden crearse a partir de archivos que guardan las configuraciones que se requiere de los mismos y los servicios que estarían utilizando, lo cual ayudaría en parte el proceso de reinstalación al momento de hacer el proceso de traslado.

### **2.2.2. Puntos en contra de la solución**

El primero punto que esta alternativa no contempla se basa en la gestión de la información y la administración de la misma, puesto que el hecho que se encuentre todo en conjunto dentro del servidor, englobando todo lo que corresponda a una unidad académica dentro de un contenedor, perdiendo el concepto del contenedor, que en esencia fundamenta el uso de una instancia por servicio, para mantener la encapsulación y modularidad de las soluciones, lo cual haría que se convirtiera en una solución más compleja y de difícil mantenimiento y acceso.

Además, se puede verificar que debido a que se tiene un solo entorno en conjunto, en dado caso alguna unidad, sobrepase el consumo de recursos podría estar tomando lo que podría llegar a necesitar otra instancia, perdiendo así la integridad de todos los sistemas que funcionan sobre el servidor dependiendo del consumo que cada uno fuera requiriendo.

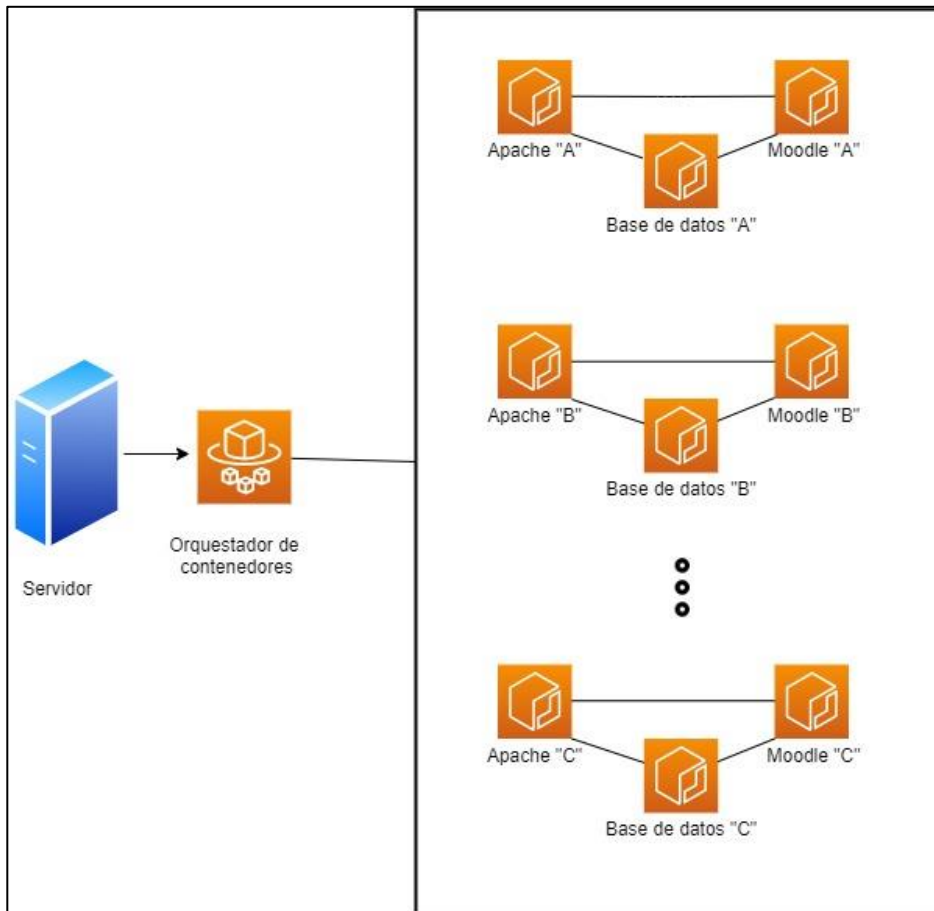
### **2.3. Instalación de un entorno de virtualización que gestione los contenedores para cada unidad académica en el servidor (entorno de virtualización para contenedores)**

La propuesta se basa en la idea de encapsular en diferentes entornos cada unidad académica, mediante un entorno de virtualización haciendo uso de contenedores de *software*, aprovechando sus características y ventajas para realizar una infraestructura modular, escalable y versátil.

Para esta propuesta de solución, se debe instalar dentro del servidor un programa informático que permita crear un entorno de virtualización mediante contenedores de *software*. Los contenedores de *software* son los encargados de alojar las diferentes herramientas que permiten la ejecución de las plataformas que utiliza cada unidad académica.

Cada contenedor de *software* debe ejecutar un único servicio, por ejemplo: un contenedor de *software* para alojar la página web, otro contenedor de *software* para ejecutar el sistema gestor de bases de datos, por mencionar algunos, para garantizar la integridad de cada contenedor.

Figura 6. Entorno de virtualización para contenedores



Fuente: elaboración propia, empleando draw.io 2000.

### 2.3.1. Puntos a favor de la solución

Se tomaron en cuenta las necesidades con que se inició el proyecto de investigación y planteó la DEDEV, de tal forma que la propuesta planteada se acople a los requerimientos de la división.



### **2.3.1.1. Integridad de los contenedores de *software***

Dedicar un único contenedor para cada servicio, disminuye la cantidad de *software* a instalar y configurar, dado que solo se instalará el *software* necesario para un único servicio: entorno de ejecución, motor de bases de datos, por ejemplo. Además, se disminuyen los problemas por dependencias de *software* evitando inconvenientes de compatibilidad o uso compartido de recursos cuando existen varios servicios que se ejecutan en una misma plataforma.

### **2.3.1.2. Versatilidad de los contenedores de *software***

Gracias a la versatilidad de los contenedores de *software* respecto a su administración: creación, modificación, portabilidad, espacio en disco, por mencionar algunas ventajas; existe la posibilidad de agregar nuevos contenedores para nuevos requerimientos, por ejemplo: agregar los servicios necesarios para agregar una nueva unidad académica; utilizar contenedores configurados con anterioridad o bien reutilizar un contenedor que tenga las mismas herramientas instaladas, disminuyendo el tiempo que conlleva realizar la configuración de nuevos contenedores.

### **2.3.2. Puntos en contra de la solución**

Se consideraron los puntos que no estaban acorde con las necesidades con que se inició el proyecto de investigación y planteó la DEDEV, de tal forma que se puedan visualizar las desventajas de la propuesta planteada.

### **2.3.2.1. Administración de recursos**

La forma de administración de recursos permitida para cada contenedor de *software* se realizan mediante línea de comandos, además, se debe realizar una monitorización constante de todos los contenedores y en consumo de recursos que realizan.

Debido al tipo de virtualización que utilizan los contenedores de *software*, se ejecutan directamente en el *kernel* del sistema anfitrión, intentando aprovechar todo el recurso de *hardware* disponible, provocando una degradación de los contenedores.

### **2.4. Instalación de un entorno de virtualización que gestione las máquinas virtuales, las cuales será una por unidad académica, dentro de las cuales se tendrán contenedores por cada servicio necesario para el sistema correspondiente (entorno de virtualización para máquinas virtuales)**

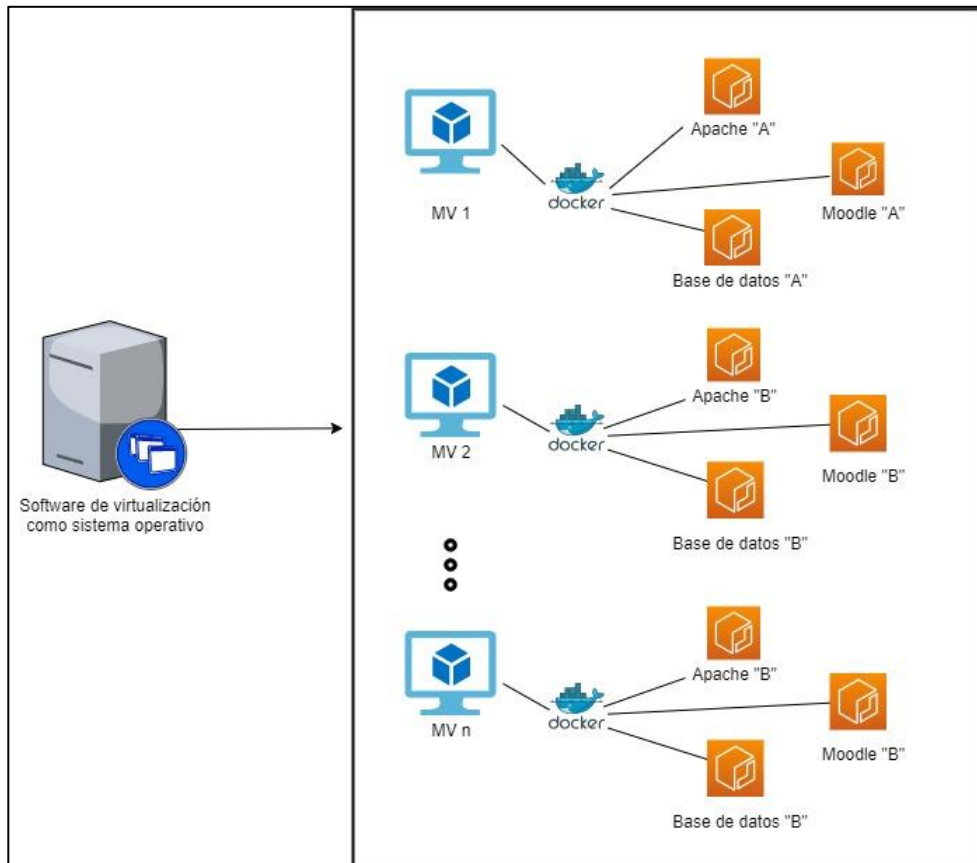
La propuesta se basa en la idea de encapsular en diferentes entornos cada unidad académica, de tal forma que estas puedan funcionar de manera independiente una de otra, no interfiriendo el tráfico y utilización de recursos de alguna en específico en todas las demás, así también se contempla que la solución de educación a distancia pueda proveerse de forma modular, siendo así más manejable y limpia su estructura.

Está basado en las ventajas que proveen las propuestas anteriores, donde trata de tomar parte de cada una de ellas, formando un ambiente mucho más completo y robusto que sea capaz de manejar de forma más ordenada y limpia la solución, tomando una máquina virtual como un entorno independiente donde

puede funcionar cada unidad académica y dentro de la cual será autoadministrado por los contenedores que serán los encargados de proveer los distintos servicios de la solución.

Se debe instalar dentro del servidor de la división un *software* que permita crear un entorno de virtualización para la distribución de recursos del servidor dentro de las distintas unidades académicas. En este entorno se crea una máquina virtual con recursos divididos de forma equitativa por cada unidad académica requerida; dentro de cada máquina virtual se debe instalar un programa informático para la administración de contenedores de *software*, cada uno de estos será el encargado de ejecutar únicamente un servicio, el cual se complementa con lo que sea provisto por los demás módulos que constituyen otros contenedores, por ejemplo: un contenedor está encargado del gestor de páginas web, otro contenedor ejecuta el motor de bases de datos.

Figura 7. Entorno de virtualización para máquinas virtuales



Fuente: elaboración propia, empleando draw.io 2000.

### 2.4.1. Puntos a favor de la solución

Se tomaron en cuenta las necesidades con que se inició el proyecto de investigación y planteó la DEDEV, de tal forma que la propuesta planteada se acople a los requerimientos de la división.

#### **2.4.1.1. Modularidad**

Dado que cada máquina virtual se asigna a una unidad académica, cada contenedor tiene asignada una única tarea para cada uno de los diferentes servicios necesarios para esta unidad académica, con esto solamente se debe instalar y configurar el *software* necesario para que dicha tarea se pueda llevar a cabo, evitando problemas de dependencias entre los distintos paquetes de *software*. Además, se garantiza que cada contenedor centrará sus recursos en la tarea asignada y evitando un impacto negativo en el rendimiento del contenedor al distribuir sus recursos en otros procesos.

Como la infraestructura de cada unidad académica es similar, se puede aprovechar cada módulo o contenedor para replicarse en nuevas unidades por configurar, ahorrando tiempo que implicaría hacer una configuración total.

#### **2.4.1.2. Administración de recursos**

Esta infraestructura permite satisfacer uno de los requerimientos principales solicitado por la DEDEV, la facilidad para administrar los recursos del servidor. Cada vez que esté calendarizada una actividad que requiera mayor uso de recursos de *hardware* en alguna unidad académica, un día de exámenes por ejemplo, el *software* de virtualización permite al administrador asignar mayor cantidad de recursos de *hardware* a las máquinas virtuales que lo requieran, los contenedores harán uso únicamente de los recursos asignados a la máquina virtual dónde estén alojados.

La solución permite que en los días en que un portal en específico necesite agregar recursos sea factible y fácil su manejo y modificación, de tal forma que incluso como se cuenta con un diseño tan robusto se pueden crear

las instancias o máquinas virtuales que se deseen si aumenta la cantidad de unidades, siempre y cuando los recursos lo permitan.

#### **2.4.1.3. Versatilidad de la solución**

Esta solución tiene la característica de poder adaptarse a lo que se plantea en la problemática ya que existe un entorno dedicado a cada unidad académica, el cual puede levantarse con facilidad en cualquier momento según los requerimientos, tal como la estructura lo define, existe una máquina virtual y dentro de esta se tiene una instancia de contenedor por servicio requerido por el sitio en cuestión, lo cual permite una propuesta más estructurada e independiente, donde la gestión de recursos de cada instancia no dependa una de otra, aun cuando estén conectados entre sí para funcionar en conjunto.

La misma versatilidad da opción a que sea un entorno cambiante y de bajo acoplamiento, lo cual da lugar a que esta propuesta sea escalable y fácil de manejar e incluso modificar según sea requerido por la DEDEV.

#### **2.4.1.4. Manejo de errores**

Como punto adicional de esta solución es que al momento de ocurrir un fallo o exista un cambio a nivel de configuración, este no afectaría todo el sistema como tal, ya que dentro de cada máquina virtual se maneja un entorno por separado e independiente, dentro del cual cada servicio que se invoca funciona de forma separada de sus contrapartes, lo cual hace más sencillo el control de este tipo de eventos.

## **2.4.2. Puntos en contra de la solución**

Se consideraron los puntos que no estaban acorde con las necesidades con que se inició el proyecto de investigación y planteó la DEDEV, de tal forma que se puedan visualizar las desventajas de la propuesta planteada.

### **2.4.2.1. Desperdicio de recursos**

Una de las principales desventajas de esta solución se encuentra en el uso y manejo de recursos del servidor anfitrión, el cual dispone de aquellos que se hayan especificado durante el contrato anual con el que cuenta la división de educación a distancia de la universidad, el cual no es un valor muy alto, por lo tanto la modularidad y encapsulamiento que supone esta implementación conlleva manejar una menor cantidad de memoria principal y secundaria, derivado a que se tiene un sistema operativo completo, con sus respectivos sistemas de archivos, tanto para el *host* como las máquinas virtuales que se tienen de cada unidad académica.

La desventaja conlleva un costo en el rendimiento y velocidad de respuesta con la que se estaría trabajando, y tendría complicaciones a menos que se tenga un método de subyugar dicho gasto de recursos en las fechas de alta demanda, que se tendrían en los días de evaluación, o trámites administrativos de los estudiantes.

### **2.4.2.2. Mayor tiempo de implementación**

Otro punto que podría identificarse como contrario en cuanto a esta propuesta se encuentra en su implementación, puesto que al ser una solución más completa y estructurada, conlleva un mayor tiempo de puesta en ejecución

debido a que debe considerarse cada una de las herramientas a instalar en el anfitrión, como el administrador de máquinas virtuales; luego la creación de las mismas, la preparación del entorno de cada una de estas, configurando su sistema operativo, variables de entorno y el administrador de contenedores; por último se debe pasar a la construcción de cada instancia por servicio requerido.

Aunque el punto es compensado, si debe tomarse en consideración que la primera implementación conlleva un mayor tiempo y esfuerzo, lo cual es un factor clave que puede ser determinante, puesto que si se tiene un equipo contratado para implementar la arquitectura y solución designada, es posible que se incurra en un gasto mayor por tener tiempos más extensos.

## **2.5. Requerimientos de solución**

La solución consiste en unificar y tratar de explotar al máximo cada una de las características que se mencionan en las opciones anteriores, con esto se busca adaptar el modelo lo máximo posible a las necesidades de negocio que se contemplan para la implementación, se debe tomar en cuenta que algunos factores determinantes para esta solución es el alto nivel de ajuste con los requerimientos solicitados por la DEDEV, de tal forma que con estas características se logra cumplir con los objetivos propuestos al iniciar el proyecto.

A partir de este punto, podría surgir la pregunta: ¿qué características se tomaron en consideración y cómo afectan o benefician al buen manejo de la estructura que se tendrá disponible para la disponibilidad de servicios y unidades académicas? razón por la cual la sección trata de contemplar cada uno de estos puntos que hacen y justifican por qué la decisión de tomar la



solución que se refirió y recomendó en su momento, por lo tanto, las características a tomar en consideración son:

El primer factor importante a destacar con respecto a las soluciones mencionadas es recalcar nuevamente que desde un inicio debido a factores de privacidad y protección de datos económicos derivado de los acuerdos en que se ven involucradas las distintas divisiones de la Universidad San Carlos de Guatemala no fue posible contar con un presupuesto o una aproximación de recursos con los que se podía implementar, únicamente se consideró el dato que no había una licitación para el uso de licencias, razón por la cual únicamente se toman en cuenta aquellas herramientas y arquitecturas que se basen en *software* libre o de licencia abierta, pero debido a no contar con ningún dato numérico que pudiera dar una idea al respecto, no es factible realizar una gestión y muestra del detalle de gastos, adicional a ello, en la solución se busca tener el enfoque más económico de tal forma que permita el buen funcionamiento con la capacidad mínima.

El entorno de virtualización, como Proxmox, desde un inicio se tomó como posible factor determinante en las distintas opciones que se plantearon debido a que en las pláticas y reuniones previas a la investigación, se consideró que para la división este *software* era una herramienta que anteriormente habían trabajado, por lo cual se solicitó si era posible la inclusión, y tomando en cuenta el hecho que debido al uso previo, la curva de aprendizaje del equipo encargado de administrar los recursos no sería tan grande, puesto que seguirán trabajando con una aplicación o entorno que conocen, por lo tanto esto incurre en menos tiempo de implementación y capacitación y ahorro de recursos económicos, por lo tanto el *gap* generado por el cambio de arquitectura no produciría un impacto a nivel de experiencia de usuario, o al menos no sería tan fuerte puesto que es una interfaz conocida.

El entorno de visualización se consideró también como parte fundamental de la solución debido a que por medio de este se provee al usuario de un medio por el cual sea posible controlar y administrar los recursos que se detallan más adelante de forma más gráfica, y adicional a ello, debido a que año tras año derivado a que se debe hacer una migración por cambio de contrato en cuanto a recursos de servidores en la nube, la tarea de implementación se vuelve tediosa y repetitiva, por lo tanto se toma en consideración el hecho que Proxmox permite realizar plantillas, que pueden ser trasladadas sin mayor inconveniente y luego de habilitar el entorno se pueden instalar de nuevo para continuar con la operación íntegra e intacta, a partir de este punto se reduce la cantidad de gastos y esfuerzo al momento de realizar la migración que se definió a realizar cada año, por lo tanto la solución propuesta cuenta con factores preventivos, lo cual es importante a corto, mediano y largo plazo, puesto que se puede reutilizar el mismo entorno, las mismas plantillas y se hace una disminución sustancial en cuanto a recurso humano y esfuerzo para cada implementación.

Las máquinas virtuales dentro de la propuesta surgieron a partir de la investigación exhaustiva que se realizó, uno de los factores más determinantes al momento de considerar el uso de máquinas virtuales más allá de contenedores directamente sobre el entorno de virtualización, es debido a que en los foros y comunidades muchos usuarios con experiencia en este tipo de arquitecturas, consideran basados en eventos previos, que es necesario contar con máquinas virtuales si se está realizando un ambiente productivo, esto derivado a que si se utilizan contenedores de forma directa puede ocasionar que el entorno sea inestable e impacte de forma sustancial en la disponibilidad e integridad de los datos que se espera estar presentando, lo cual podría ocasionar molestias e incluso la desconfianza de los alumnos o usuarios finales que consuman los servicios que se espera levantar sobre esta arquitectura, así

que debido a ello se consideró el colocar máquinas virtuales que permitan tener una base más sólida y estable.

Otra razón que influye en el uso de máquinas virtuales es debido a la organización que se pretende pueda tener el espacio dedicado a cada unidad académica, considerando lo expuesto por la DEDEV se tomó en cuenta que existen múltiples unidades que funcionan de forma paralela, razón por la cual es necesario el gestionar qué cantidad de recursos se estará dedicando a cada una de ellas, lo cual puede hacerse por medio de un análisis de objetivo, donde se presente el consumo y población con que cuenta cada una de ellas, de tal forma que desde Proxmox sea factible el gestionar qué cantidad será proporcional en cada máquina virtual, que cada una de estas debería de considerarse como unidad académica, de tal forma que exista una relación de uno a uno entre estos dos elementos, así, se logra una administración más ordenada y certera, además que permite un mayor control sobre la ejecución de los servicios.

El uso de contenedores no se deja por fuera de la solución, puesto que la propuesta contempla que estos se generen para los servicios que se estarán implementando sobre cada máquina virtual, ahora, ¿porque es necesario esto? podría parecer redundante el utilizar contenedores dentro de máquinas virtuales, puesto que en teoría estos buscan minimizar la cantidad de recursos que impondría una máquina con su sistema operativo huésped, sistema de archivos y demás elementos que consumen recursos y no están dedicados al foco principal que es presentar sistemas de Moodle a los alumnos.

La razón por la cual no se utilizaron directamente los contenedores sobre el gestor se explicaba anteriormente que se deriva de la estabilidad que busca la propuesta, pero dentro de cada máquina virtual puede explotarse los

beneficios de su uso, entre ellos la modularidad que permite el tener una aplicación sobre contenedores, lo cual ayuda a que sea más fácil la gestión y algún escalamiento en dado caso se requiera de esta forma, así mismo se puede separar la base de datos del servicio para los mantenimientos que puedan presentarse en un futuro.

Otro punto que se toma en cuenta en el uso de contenedores es el hecho que dentro de las máquinas virtuales se espera no tener que preocuparse por la gestión de espacio o recursos que necesite cada instancia, puesto que el contenedor por sí mismo puede hacerlo y de esta manera toda la unidad académica tiene una autogestión que permite que no sea necesario estar completamente dedicados al comportamiento de los servicios instalados sobre está, más bien, se pretende que la carga y esfuerzo a este nivel no se presente y sea controlable desde afuera cualquier movimiento o dato que se necesite, cuestión que de igual forma se ajusta al uso de contenedores ya que estos permiten un direccionamiento de puertos sobre la máquina anfitriona, así como de carpetas, lo cual ayuda a que la información sea accesible y esté protegida ante cualquier evento desafortunado.

Como contenedores independientes se considera que se tenga las bases de datos que correspondan, otro para el servicio de Moodle y otro para apache, de tal forma que se preserve la modularidad y sea más fácil el uso y mantenimiento así como manejo de errores en caso existan dentro de la arquitectura que se presenta, con un formato modular es posible identificar más rápido los inconvenientes que puedan afectar a los estudiantes que hagan uso de la herramienta.

### **3. SOLUCIÓN PROPUESTA**

Es importante identificar las herramientas que mejor utilizan los recursos con los que se dispone, ya que el objetivo es tratar de explotar al máximo las diferentes soluciones que se proponen en el capítulo anterior, tomando de base que para cada elemento que se requiere se tiene más de una opción con diferentes características que pueden ser utilizadas para potenciar cada una de las integraciones que se mencionaba anteriormente, sacando de ellas todos los puntos a favor y en contra que se tienen de las soluciones estudiadas.

Este análisis nos permite tener una visión más objetiva de lo que nos puede proveer cada opción, de tal forma que sea más fácil de identificar aquella que se adecue más a las necesidades que se tiene en la división, puesto que lo que se busca es llevar un producto que sea capaz de cumplir con los requerimientos, tomando de base los sistemas, o servidores con que se cuenta actualmente.

#### **3.1. Plataforma para gestión y creación de contenedores**

En el caso de este punto, es importante destacar que una de las más grandes razones por las cuales se decidió por un *software* en específico es debido a la popularidad de este, debido a que es un factor importante al momento de necesitar algún tipo de soporte, puesto que si es una herramienta utilizada a nivel mundial por una gran comunidad, la documentación y el registro de posibles errores y sus respectivas soluciones es mucho más fácil encontrar, por lo tanto una herramienta con tal cantidad de usuarios que permite encontrar cualquier incidente es en definitiva la solución correcta que puede elegirse,

debido a que se prevé situaciones adversas para el futuro o posibles mantenimientos y el debido escalamiento que podría sufrir el sistema a lo largo del tiempo.

Por lo tanto, se eligió Docker puesto que entre las diferentes opciones con que se contaba al inicio, esta es la solución que se asemeja más a lo que se buscaba con respecto a los requerimientos, así como la cantidad de contenido en internet que se visualiza con respecto a este *software* y el soporte de diversas comunidades que respaldan su uso, por lo tanto permite tener un manejo más amplio y coordinado de la información que se pretende almacenar.

Debido a que se toma este *software* como el definitivo para la solución propuesta se amplía el conocimiento general que previamente se expuso en un capítulo anterior, detallando más a fondo lo que esta herramienta puede ayudar a realizar, tomando en cuenta que las demás que se analizaron son opciones muy viables y con muchas ventajas que se consideraron a fondo pero debido a que se contaba con un requerimiento ajustado se eligió esta plataforma.

### **3.1.1. ¿Por qué utilizar contenedores?**

Este tipo de herramientas son posibles debido al aislamiento y la virtualización que permiten sobre procesos del sistema operativo, donde es posible que diferentes componentes de alguna aplicación compartan los recursos que provee el *kernel* del sistema operativo anfitrión, por lo tanto se podría deducir que comparte los beneficios de una máquina virtual, como lo es que se tiene un aislamiento de aplicaciones, escalabilidad rentable y disponibilidad, pero adicionalmente puede observarse un nivel de abstracción que permite ventajas como lo son las siguientes.

Primero es importante identificar una productividad mejorada para las personas que harán el desarrollo del sistema en comparación con su antecedente, las máquinas virtuales, puesto que es mucho más rápido y fácil de aprovisionar, implementar y reiniciar, eso es un factor fundamental para la canalización de una integración y entrega continua, lo cual hace de ello una característica ideal para adoptarse en prácticas ágiles y DevOps.

Luego basándose en la cantidad limitada de recursos con que se cuenta, se puede valer de que este tipo de soluciones hace un mejor uso de ello, provocando una eficiencia mayor, puesto que puede ejecutar gran cantidad de veces copias de una aplicación dentro del mismo *hardware*, y por último es importante destacar la ligereza con la que cuentan, puesto que cada instancia no lleva la carga útil de un sistema operativo, solamente los procesos y dependencias de este que sean necesarios para la ejecución del código que quiere resguardarse dentro del mismo.

#### **3.1.1.1. Definición de Docker**

Es una plataforma de contenerización código abierto para la creación, implementación y administración de aplicaciones en contenedores que pertenece al proyecto de *software* libre perteneciente a la empresa Docker Inc, el cual permite a las personas que estén implementando el sistema empaquetar las aplicaciones que desean tener disponibles en componentes ejecutables de forma estandarizada, de tal forma que se pueda combinar el código fuente con las bibliotecas o dependencias necesarias, de tal forma que este sea un entorno totalmente independiente y pueda conectarse a cualquier sistema.

Es importante destacar que realmente es posible realizar contenedores sin hacer uso de una herramienta como Docker, pero este permite que ese diseño

y construcción sea mucho más fácil, seguro y sencillo, de igual forma da una solución al trabajo que requeriría la implementación y administración de estas soluciones, debido a que consta esencialmente de un conjunto de herramientas en la cual estas tareas anteriormente descritas se pueden realizar de forma simple y automatizada.

#### **3.1.1.1.1. Uso de Docker**

Un punto imprescindible es la popularidad con que cuenta y el hecho que los contenedores se utilizan indistintamente, cabe destacar que esta herramienta no es la primera que salió al mercado con la idea de empaquetar las aplicaciones en módulos independientes, puesto que el primero en hacerlo antes que Docker en el 2013 fuera lanzado, fue LXC que se hizo notar desde el 2008 implementado en un *kernel* Linux, lo cual restringía únicamente este sistema operativo para la virtualización, pero Docker aprovechó al máximo la solución que provee LXC y añadiendo luego de algunas versiones a su propia tecnología personalizadas nuevas características.

La versión de Docker incluyó en su sistema un peso más liviano y actualizaciones más granulares, debido a que solo se puede ejecutar un proceso en cada contenedor, es posible que una aplicación pueda seguir funcionando mientras se realizan reparaciones o actualizaciones de algún módulo en caliente, además permite una portabilidad mejorada libre de problemas, superando por completo a su antecesor LXC quien hace referencia frecuentemente a configuraciones específicas en la máquina, los contenedores que se realizan en Docker se pueden ejecutar sin modificaciones independiente del entorno de escritorio, nube o centro de datos en el que estén habilitados.



También la administración se ve claramente mejorada puesto que primordialmente se cuenta con una creación de contenedores automatizada, donde Docker puede construir el contenedor únicamente con el código fuente de la aplicación de forma automática, esto acompañado de un control de versiones donde la herramienta permite el rastreo de versiones de una imagen donde puede administrarse y cargarse los deltas necesarios entre una versión activa y una nueva.

Las características anteriores asociadas con la distribución y gestión mejorada debido a la capacidad de reutilización de contenedores existentes que pueden utilizarse como imágenes base, se ajusta al requerimiento inicial que se tenía de esta solución de contar con plantillas para las recurrentes migraciones que año con año se realizan y las bibliotecas de contenedores compartidos permiten que se pueda acceder a un registro de miles de colaboradores que aportan y contribuyen usuarios de este *software* haciendo que no sea necesario realizar la construcción desde cero.

Esta personalización que logró Docker ha hecho que desde su adopción se haya disparado de forma inmediata y aun siga aumentando, contando con el respaldo de 105 mil millones de descargas de contenedores, por lo tanto el soporte y comunidades que contribuyen frecuentemente al mejoramiento y solución de incidentes de esta herramienta lo hacen ideal para la implementación de un sistema que se desea que sea lo más automatizado y fácil de administrar posible.

### **3.1.2. Hipervisor**

Esta propuesta se tiene como una sugerencia con el enfoque de adaptar la solución lo máximo posible al modelo que actualmente se cuenta, de tal

forma que el impacto que este genere dentro de la solución no encuentre discrepancias con la idea que cuentan las personas que administran el mismo, para no contar con diferencias sociales o de entendimiento de la metodología utilizada, además, se toma en consideración debido a que a partir de los hipervisores se logra tener un mayor control y dominio de cada unidad académica por medio de las máquinas virtuales, de tal forma que sea factible para el administrador tener cierta influencia en la solución que se marca para puntos de inflexión, donde exista mayor demanda para ciertas divisiones, debido a época de evaluaciones o actividades específicas donde podrían necesitar un mayor uso de recursos, que aunque la solución cuenta con una administración automatizada, se desea contar con un límite impuesto que permita que en algún momento se consuma lo que se ha asignado en otros incidentes o necesidades que se puedan tener a lo largo del uso de la solución.

### **3.1.2.1. Definición y funcionamiento de Proxmox Virtual Environment como hipervisor**

También conocido por su abreviatura Proxmox VE, es una alternativa entre plataformas empresariales para administrar la virtualización de servidores, de código abierto o de colaboración abierta. Está basado en el sistema operativo Debian GNU/Linux, pero posee su propio *Kernel* personalizado; como se mencionó con anterioridad, al ser un software de código abierto, el código fuente de Proxmox VE es gratuito y su lanzamiento se hace bajo la licencia GNU Affero General Public License, v3 (GNU AGPL, v3), con esto se obtienen beneficios como utilizar el *software* de forma gratuita y algunos otros beneficios descritos a detalle posteriormente.

Proxmox VE posee dos tecnologías primordiales, que permiten realizar dos tipos de virtualización: Virtualización completa mediante KVM o *Kernel-*

*based* Virtual Machine y su traducción es “máquina virtual basada en el *kernel* o núcleo”; y virtualización basada en contenedores mediante contenedores Linux o por sus siglas en inglés, LXC, LinuX Containers.

Además, se puede definir almacenamiento definido por software, configuración de los requerimientos funcionales de las redes para las máquinas virtuales dentro de la misma plataforma, por mencionar algunas de las características más relevantes de Proxmox.

#### **3.1.2.1.1. Uso de Proxmox Virtual Environment Virtualización mediante KVM**

De sus siglas en inglés *Kernel-based* Virtual Machine “Máquina virtual basada en el *Kernel*”, es una tecnología de código abierto para la virtualización completa para sistemas operativos Linux, permitiendo a estos sistemas operativos en hipervisores ejecutar diferentes entornos de virtualización o máquinas virtuales utilizando imágenes de sistemas operativos Linux o Window. KVM consiste en un módulo que se puede cargar en el *Kernel*, que provee una infraestructura de virtualización central y un módulo específico para procesadores Intel o AMD. Cada máquina virtual posee su *hardware* privado, por ejemplo: una tarjeta de red, disco para almacenamiento, tarjeta gráfica, por mencionar algunos ejemplos.

KVM viene incluido en sistemas operativos Linux desde las versiones 2.6.20 o posteriores, permitiendo aprovechar todas sus funcionalidades al poseer una versión de estos sistemas operativos.

- Licencia *Open Source*:
  - Cualquier *software* de código abierto *u open-source*, por su término en inglés, permite a cualquier persona utilizar dicho programa.
  - El proyecto es mantenido por una comunidad de desarrolladores, que está en constante desarrollo de mejoras, arreglo de fallos actuales, apoyo y resolución de dudas para los usuarios del programa a través de comunidades que se manejan mediante diferentes foros.
  - Proxmox VE posee una licencia de código abierto, por lo cual el usuario tiene acceso total a cualquier funcionalidad de este software, de forma gratuita, así como un alto nivel de seguridad y confiabilidad; también se tiene acceso al código fuente para poder realizar contribuciones al proyecto.

### **3.1.3. Plataforma de gestión de virtualización**

El motivo principal del uso de una Plataforma de Gestión de Virtualización es centralizar la administración de todas las tareas en una sola herramienta, reduciendo el esfuerzo y complicaciones que conlleva tener diferentes herramientas para las diferentes tareas involucradas al administrar máquinas virtuales, un sistema operativo anfitrión, una interfaz de usuario, un hipervisor, un administrador para el almacenamiento, la compatibilidad y actualización del software de las diferentes herramientas; por mencionar algunos inconvenientes que esto conlleva.

Al centralizar esta administración, la herramienta posee diferentes módulos que permiten llevar a cabo diferentes tareas de forma sencilla a través de una interfaz gráfica de usuario. No existe incompatibilidad entre los diferentes programas que incluye la plataforma de gestión de virtualización, y no existe inconveniente al actualizar dicho *software*.

### **3.1.3.1. Definición y funcionamiento de Proxmox Virtual Environment como plataforma de gestión de virtualización**

Proxmox VE es una plataforma completa, de código abierto para virtualización empresarial, integrando dos diferentes tecnologías para la virtualización, manejo de almacenamiento y redes, clúster de alta disponibilidad, herramienta para la recuperación de desastres, todo a través de una interfaz de usuario de fácil uso.

Al poseer dos tecnologías para la virtualización permite adaptarse de manera más flexible a las necesidades del negocio para cualquier entorno de producción.

El soporte empresarial está disponible a través de una suscripción de pago, pero se tiene la opción de acceder sin costo a los foros y comunidad de Proxmox VE para consultar fallas y errores comunes.

- Administración centralizada:

Proxmox VE permite realizar la administración de todas las tareas a través de una interfaz gráfica de administración centralizada, que es muy intuitiva de utilizar y amigable con el usuario, evitando el uso e instalación de

herramientas externas para dichas tareas. En esta interfaz se realizan operaciones desde el despliegue de máquinas virtuales o contenedores hasta el manejo de clúster de nodos de máquinas virtuales.

La GUI o interfaz gráfica de usuario, por sus siglas en inglés Graphic User Interface, está basada en el framework ExtJS de JavaScript, por lo que se puede acceder desde cualquier navegador web actual.

Otras tareas importantes a destacar que se realizan desde la interfaz son tareas como las migraciones “en vivo” o “en línea” de las máquinas virtuales, a través del uso de *script* o mediante la interfaz incluida en Proxmox VE hacia otro servidor o dentro del mismo servidor para clonar máquinas virtuales existentes y poderlas reutilizar, ejecutar tareas para el respaldo de datos, la administración de diferentes clústers de forma central sin necesidad de utilizar una herramienta para cada clúster; entre otras funcionalidades por destacar. Además, se puede visualizar el historial de tareas realizadas y registros del sistema.

- Sistema de archivos:

Proxmox VE posee su propio sistema de archivos desarrollado por el equipo de Proxmox, dicho sistema de archivos se llama ‘pmxcfs’ por sus siglas en inglés de “Proxmox Cluster File System”. Este sistema de archivos permite sincronizar los archivos necesarios para las configuraciones a través de los diferentes clúster, replicando estos archivos dentro de todos los clúster disponibles. Toda esta información se almacena en una base de datos en disco, al igual que una copia se almacena en la memoria RAM.

- Sistema migración:

Proxmox VE posee una característica para la migración “en vivo” o “en línea” para mover un nodo de clúster hacia otro, sin que el usuario final se percate de este proceso. Este proceso se puede realizar desde la interfaz de usuario o desde la línea de comandos, reduciendo los tiempos que el sistema se encuentra apagado para realizar algún tipo de mantenimiento.

- Interfaz de línea de comandos:

Por sus siglas en inglés “*Command Line Interface*” o CLI, es una herramienta para usuarios avanzados que están familiarizados con otras herramientas similares como la *shell* de Unix o Windows Powershell.

- Administración basada en roles:

Proxmox VE permite el acceso a cada componente de la herramienta, como las máquinas virtuales, el almacenamiento, los nodos de clúster, por ejemplo; mediante un sistema de administración de permisos basado en roles. Un administrador general puede delegar ciertos permisos y controlar el acceso a ciertos componentes a diferentes usuarios, también conocido como “Lista de control de accesos”.

#### **3.1.4. Sistema operativo recomendado**

Dado que Proxmox VE es una distribución de Linux basada en Debian, la instalación se puede realizar de dos formas distintas, ambas usando el sistema operativo mencionado.

### 3.1.5. Instalación de Proxmox utilizando Proxmox VE Installer

La instalación de Proxmox VE utilizando el instalador que ellos proveen, es la forma recomendada cuando se dispone del acceso al servidor para poder realizar una instalación completa del sistema operativo, ya que existen proveedores de servicios en la nube que solamente dan al usuario máquinas virtuales con sistemas operativos instalados.

La imagen ISO de Proxmox VE Installer incluye un sistema operativo Debian y todos los paquetes necesarios para Proxmox VE: el instalador de Proxmox VE para particionar las unidades de almacenamiento en los formatos soportados, el *Kernel* Proxmox VE Linux para el soporte de máquinas virtuales mediante KVM y contenedores de software mediante LXC, las herramientas necesarias para la administración de máquinas virtuales, contenedores, el sistema anfitrión, clústeres y todos los recursos del sistema, y la interfaz web para realizar dicha administración.

La instalación de Proxmox VE se realiza a través de una interfaz gráfica, por lo que este método se recomienda para usuarios con un conocimiento intermedio de sistemas operativos Linux, ya que se realizan configuraciones como el particionamiento de las unidades de almacenamiento, el tipo de sistema de archivos que utilizará el sistema operativo, por mencionar algunas configuraciones.

Los recursos para realizar esta instalación están disponibles en el sitio web de Proxmox VE, así como videos en la plataforma de videos, Youtube.



### **3.1.6. Instalación de Proxmox sobre Debian Buster**

El método alternativo para esta instalación, se realiza directamente en un sistema Debian previamente instalado en servidor, pero es recomendado para usuarios avanzados.

Se debe contar con una instalación previa del sistema operativo Debian Buster, para casos en los que se desea utilizar un diseño de particiones de las unidades de almacenamiento personalizado.

Se recomienda esta instalación para usuarios con conocimientos avanzados en sistemas operativos Linux, ya que se deben realizar configuraciones de redes, agregar repositorios para la descarga de Proxmox VE, instalar los paquetes de Proxmox VE, eliminar algunos paquetes que ya no son necesarios, conectarse manualmente a la interfaz de Proxmox VE, opcionalmente remover el Kernel de Debian y actualizar el grub, entre otras configuraciones.

### **3.2. Impacto social y de desarrollo a nivel nacional de la propuesta de solución**

La organización Mundial de la Salud, derivado a la situación sanitaria que se ha desarrollado a causa de la pandemia SarsCov2, conocida popularmente como Covid 19, ha recomendado que se tomen medidas para la prevención de contagios, mayormente masivos, entre los cuales destaca entre ellos el distanciamiento social, acompañado de recomendaciones de higiene (como el lavado de manos, uso de mascarilla, entre otros) y constante control de salud para que se pueda mitigar en gran medida el riesgo que supone de forma global esta enfermedad.

Acatando las instrucciones impartidas cada gobierno y entidad ha colocado ciertos protocolos que se ajustan a las necesidades comunitarias, de esta forma pueden seguir operando sin tener que detener el crecimiento a nivel global en las distintas áreas que correspondan.

En marzo de 2020 se dio a conocer a los medios, que el rector en funciones dejaba suspendida las actividades académicas de la Universidad San Carlos de Guatemala, con efecto inmediato a partir del siguiente día por la premura con que se contaba debido a que se había anunciado el primer caso de contagio dentro del país, lo cual inicialmente había sido dispuesto con una duración de un mes, siendo vigente hasta el 14 de abril del mismo año, lo cual no pudo ser posible debido a la gravedad con la que se disparaban los contagios para esas fechas.

La necesidad de tener una solución con la cual se pudiera administrar de mejor forma los recursos con que cuentan las distintas unidades de la División de Educación a Distancia (DEDEV) para suministrar las herramientas con que los alumnos acceden al contenido de sus respectivas carreras se hizo evidente, puesto que no se podía parar la educación a nivel superior en el país, puesto que esto ocasiona un retraso en el progreso nacional y seguramente una disminución de la economía. El porcentaje a nivel nacional de estudiantes universitarios según un censo realizado el año 2019, previo a la pandemia actual muestra una tasa del 4,56 por ciento de la población que tiene acceso a estudios de pregrado, y un 0,48 por ciento, lo correspondiente a postgrados, desde los incidentes ocurridos debido a las mismas circunstancias sanitarias, dichos números han bajado pero, la realidad es que una gran población ha logrado mantenerse activa académicamente por los medios virtuales que se han provisto y plataformas de comunicación que se han abierto a partir de las necesidades.

Todas las circunstancias descritas llevaron a la razón por la cual se decidió realizar un estudio en el cual se plantea una solución de administración de los recursos, herramientas y plataformas de las distintas unidades de la División de Educación a Distancia (DEDEV), pero que la misma se ajustara al modelo de capacidad con el que se cuenta, puesto que no es factible aumentar este, sino únicamente modificar su estructura de tal forma que se pueda aprovechar al máximo, de tal forma que se cuente con aplicaciones robustas que puedan funcionar sin problema y sea administrable para mitigar incidentes en épocas pico donde pudiera tenerse un mayor consumo desde cierto punto.



## **4. DISEÑO Y PROPUESTA PARA IMPLEMENTACIÓN DE SEGURIDAD**

### **4.1. Diseño de seguridad**

La seguridad es uno de los conceptos más importantes a considerar en este capítulo ya que juega un papel muy importante dentro de la informática, y uno de los objetivos es proteger la integridad y la privacidad de la información, por ello es sumamente importante contar con todas las medidas de mitigación ante ataques cibernéticos que pueden llevar a el robo de la información, encriptación de ficheros, instalación de programas que puedan estar en constante ataque a la información que encuentre en el ordenador; todo esto por personas ajenas que buscan filtrar datos que sean de su interés. Por lo tanto se debe estar preparados y capacitados ante cualquier amenaza para poder hacer uso de los recursos de la mejor manera posible, evitando así, los ataques constantes que se hacen hoy en día.

Para poder tener una buena práctica de la seguridad, se tienen varios caminos, ya que esto depende de la lógica que se use dentro de una empresa u organización para mantener la integridad, confidencialidad, disponibilidad y privacidad de un sistema que contenga la información; algunas de las herramientas que se utilizan o según nuestro criterio se deberían utilizar en empresas u organizaciones para el resguardo de la información son:

- La encriptación de la información: esta es una buena práctica que se ha utilizado desde hace mucho tiempo. Algunos ejemplos describen que se ha utilizado la criptografía en los primeros siglos antes de Cristo, por lo

que es una técnica muy utilizada hoy en día, que fue creada con el fin de preservar la privacidad y evitar que una persona no autorizada robe la información que se está transmitiendo, un claro ejemplo de esto es la aplicación de mensajería WhatsApp que busca mantener la privacidad de los mensajes que viajan de un destinatario a otro, manteniendo así el resguardo y la seguridad de que no se va a perder o que esta información caiga en manos ajenas.

- Programas de antivirus: esta herramienta es un muy efectiva ya que permite contar con las medidas de protección ante la detección de *malware* que esté intentando quedarse en el ordenador ya que, cualquier dispositivo que esté conectado en la red puede sufrir este tipo de ataques si está vulnerable y no cuenta con un programa que evite el paso de estos programas maliciosos, es por ello que es recomendable contar con un programa que alerte de los posibles ataques a la información personal.
  
- Servidor proxy: utilizar un servidor proxy dentro de la organización hace que sea una medida más de protección, ya que uno de los enfoques más importantes que se le dan a esta herramienta es que sirve como intermediario entre el servidor local y la nube, que en este caso es la salida a internet. Algunas de las características con las que cuenta este servidor proxy son:
  - Control de acceso: por tratarse de un servidor de la empresa, existe un administrador de la red y este puede tener el control de poder otorgar o denegar permisos a usuarios en específico, IP's en específico e incluso segmentos de red, colocando restricciones para que dichos usuarios tengan o no acceso a internet.

- Filtrado de contenido: en el caso de esta característica se puede permitir el acceso o no a ciertos sitios web, colocando reglas que impidan ver su contenido, ayudando así a tener una capa más de seguridad y un mejor control de lo que se busca en internet y evitar ser víctima de ataques por los ciberdelincuentes.

Así como estas herramientas que se enfocan en algo en específico existen muchas más que hoy en día utilizan las empresas aplicando este tipo de estrategias en sus arquitecturas y que es de vital importancia tener el conocimiento y estar capacitados para hacer el mejor uso de las mismas.

Cabe resaltar que para este capítulo se busca dar un enfoque de seguridad a la propuesta de la solución discutida en capítulos anteriores y para ello se tienen los siguientes conceptos que son importantes y que deberían de considerarse para su futura implementación.

#### **4.1.1. Seguridad en la Nube**

Este concepto de la “nube” engloba tantos términos y usos que se le pueden dar, pero para fines de este capítulo no se va a entrar en tanto detalle, sino que se va a dividir en dos grupos que se van a enfocar en la seguridad que va a otorgar el proveedor de servicios y la seguridad que debe tener la aplicación del cliente, porque no se puede dejar toda la tarea de un solo lado ya que existe la posibilidad que el atacante burle la seguridad del proveedor de servicios pero se tope con otro muro de herramientas que evite el robo de la información del cliente y por ello es importante mencionarlos.

- Proveedor de servicios y cliente adquiriendo un servicio

De un proveedor de servicios en la nube es importante conocer las tecnologías, la infraestructura con la que se maneja los recursos y que es lo que ofrecen al cliente, ya que se está almacenando toda la información de una empresa u organización y por ello es importante resaltar esta parte para que se estudie cuáles son las herramientas o aplicaciones que ellos ponen de prioridad para poder resguardar los datos de toda una compañía, ya que como clientes que están adquiriendo el servicio se tiene esa ligera desconfianza por presentar información sensible ante un proveedor que posiblemente es la primera vez que se está obteniendo un servicio como tal y por ello cabe destacar que desde un inicio debe haber transparencia en los acuerdos y políticas que se le presentan al cliente en donde se presentan la confidencialidad y hacen mención de cómo mitigar problemas potenciales en caso de presentarse alguno de estos.

Algunas de las características que como clientes se debe solicitar o buscar que un proveedor de servicios tenga es que ofrezcan servicios de redes privadas, que incluyan la administración de firewalls, sistemas de detección de intrusos, análisis de vulnerabilidades, soluciones contra códigos maliciosos, respaldo y recuperación de la información, son aspectos que se deben considerar al momento de adquirir o solicitar un espacio a un proveedor de servicios.

Así mismo, como clientes adquiriendo un servicio se pretende tener disponible en todo momento la información y para ello se solicita tener acceso desde cualquier lugar y en cualquier momento, también algo que el cliente debe considerar es verificar si su proveedor cuenta con flexibilidad; es decir cómo pueden escalar los servicios en dado caso aumenten o disminuyan las necesidades de la compañía. Estos temas a considerar no representan un



problema para los proveedores, ya que hoy en día cualquiera otorga tantas ventajas para poder adaptarse a las necesidades de los clientes, haciendo esto mucho más atractivo para las empresas u organizaciones que desean adquirir estos servicios.

¿Qué tan vulnerable se puede estar al contratar un alojamiento en la web con un proveedor de servicios?

Esta es una de las preguntas que muchos se deben hacer para poder tener presentes qué vulnerabilidades se encuentran en la web y cómo pretenden mitigarlas nuestro proveedor de servicios, entre las más conocidas o más comunes se puede encontrar:

- Configuración del servidor Web: aunque pareciera algo en lo que la empresa no debiera preocuparse, al contrario es muy importante contar con las configuraciones adecuadas del servidor donde se tiene alojados los sitios web ya que es común que aquellos que cuentan con alojamientos compartidos no se percatan de la configuración de los servidores, ya que lo dejan todo en manos del proveedor del servicio porque es normal que uno adquiera un servicio y ellos se encarguen del resto, pero se necesita tener conocimiento de por lo menos las configuraciones básicas, como por ejemplo que se cuente con las actualizaciones de aplicaciones que están en ejecución ya que una aplicación desactualizada da lugar a que se mantenga fallando, así mismo configuraciones básicas de seguridad para que queden correctamente y no exista una filtración por parte de los atacantes al momento de realizar alguna autenticación.

- Vulnerabilidades de *software* y *malware*: aunque va de la mano con la configuración del servidor web, ya que esta solicita que se tengan las aplicaciones actualizadas y de igual forma corren riesgo ya que pueden existir programas maliciosos en el alojamiento compartido que de una u otra forma van a llegar a afectar eventualmente.
- IP's compartidas: esto es algo de lo que las empresas o los clientes deben estar enterados al momento de adquirir un servicio, se debe preguntar por este tema y no permitir que el proveedor de servicios le venda la idea que es una buena idea, ya que existe un alto impacto que sufran algún ataque no directamente a su sitio web sino el del vecino, pero solo con el hecho de compartir IP ya queda una serie de ataques que los ciberdelincuentes pueden realizar y dejar vulnerable cualquier sitio que esté alojado en su espacio en la nube, por lo tanto se debe evitar adquirir este punto.

#### **4.1.2. Seguridad en máquinas virtuales**

Hablar de máquina virtual es como hablar de un servidor físico por lo tanto se va a buscar tener las mismas configuraciones de seguridad para este tipo de servidor virtual ya que se utiliza una instancia de sistema operativo y se trata como un servidor físico la diferencia radica en que no se cuenta con él físicamente pero puede llegar a tener los mismos recursos y las mismas especificaciones que el usuario quiera, algunas de las recomendaciones para evitar cualquier tipo de ataque a la máquina virtual son las siguientes:

- Uso de consola: está claro que tanto en un servidor virtual como en uno físico se cuenta con el acceso a la consola del sistema y esto es un tema

delicado ya que al tener este control puede cualquier persona ingresar a la administración de la consola y realizar configuraciones que afectan a la máquina y al sistema operativo por lo tanto esto lo hace vulnerable y permite que cualquier ataque malicioso pueda ocurrir en dicha máquina por lo tanto una de las recomendaciones para este tipo de vulnerabilidad es minimizar el uso de la consola de la máquina virtual o limitar el acceso.

- **Uso de Plantillas:** es muy común ver que al momento de crear una máquina virtual se realice de forma manual ya que es muy sencillo de realizarlo porque solo se debe tener la imagen del sistema operativo y conocer un poco sobre las configuraciones de dicho sistema y se instala, sin embargo esto puede llegar a ser muy arriesgado cuando se trata de una máquina virtual que contendrá información delicada ya que para estos casos se recomienda utilizar plantillas para implementar máquinas virtuales que vengan con un nivel de línea base conocido de seguridad, también que contenga un sistema operativo protegido y tenga las configuraciones adecuadas para lo que se vaya a utilizar. Además estas cuentan con otra ventaja ya que pueden levantarse todas las instancias que se deseen con las configuraciones necesarias y/o adaptadas al usuario.
- **Evitar control de recursos:** es una de las recomendaciones que hay que tomar en cuenta ya que el mal uso de los recursos en una máquina virtual puede provocar una denegación de servicio más conocida como DoS por lo tanto para evitar esto se recomienda que se utilice la características de administración del host, así mismo se debe otorgar CPU y memoria suficientes para que funcione de forma adecuada utilizando los recursos compartidos asegurando que dichas máquinas

virtuales están agrupadas para que tengan recursos similares con esto se evita que una máquina virtual utilice más recursos de los necesarios. Con estos límites de recursos se protege a la máquina virtual de otras que tengan un consumo alto de recursos de hardware compartido.

#### **4.1.3. Seguridad en contenedores**

Los contenedores permiten empaquetar y diseñar aplicaciones en diferentes entornos, facilitan la forma de levantar un servicio o una configuración ya que están basados en imágenes o bien un archivo que contiene toda la información necesaria que ejecuta el contenedor, pero esto como cualquier otro sistema debe contar con las medidas de seguridad, existen varias prácticas que pueden ser útiles y que es importante mencionar para poder hacer uso de estas y seguir las recomendaciones que se tiene para evitar tanto ataques como robo de información, algunas de ellas son:

- Recursos en los containers: el uso compartido de un mismo sistema operativo puede ser el punto vulnerable y el atacante aproveche para poder adueñarse de los recursos del sistema, por ello es importante hacer uso correcto de las herramientas para poder tener en constante monitoreo nuestro contenedor entre estas se destacan el Namespaces y el Cgroups que su principal función es limitar la cantidad tanto de CPU como de memoria que se le puede otorgar a cualquier contenedor.
- Automatización de la implementación: se considera una buena práctica de seguridad en contenedores cuando se identifica un problema y en este no se ejecutan parches para solventarlo sino que se vuelva a hacer el diseño del contenedor. Es por ello que es muy importante entender cómo automatizar las políticas para poder identificar los diseños que

pueden llegar a tener algún problema de seguridad, en este punto la clave es saber identificar las políticas que permiten realizar el rediseño automático de los contenedores.

- Utilizar herramienta de seguridad: esta práctica va de la mano de la mencionada anteriormente ya que para poder contar con una herramienta de seguridad que esté monitoreando cada contenedor se vuelve una tarea un poco complicada ya que en cada uno de estos se puede contar con múltiples paquetes a la vez es por ello que se menciona en la práctica anterior que se debe contar con una automatización al realizar este proceso el análisis de vulnerabilidades en los contenedores ya que esto sirve para poder mantenerte alerta por lo tanto se debe buscar o contar con una herramienta que sea eficiente y que pueda detectar vulnerabilidades en tiempo real y que se enfoque específicamente en el desarrollo y la administración de las aplicaciones.

#### **4.2. Diseño de la arquitectura**

La seguridad es uno de los papeles más importantes a considerar para la implementación de este capítulo, ya que se debe contar con buenas prácticas para poder realizar el diseño de una arquitectura que sea robusta ante cualquier ataque y que cuente con la mayor cantidad de herramientas posibles para mitigar cualquier intento de robo de información, código malicioso, encriptación de la información, entre otras que fueron mencionadas en secciones anteriores. Es por ello que esta sección se detalla sobre las herramientas que se deben tener para diseñar una arquitectura que contemple las buenas prácticas que se han mencionado en este trabajo de investigación, esta será una propuesta para la división de educación a distancia en entornos virtuales y que es importante que contemplen una parte de lo que se debería solicitar al momento de adquirir

un servicio con algún proveedor; así mismo este diseño servirá para implementar en su división un nivel de seguridad mayor al que posiblemente cuentan actualmente.

En este capítulo se detalla la arquitectura propuesta, la cual se divide en dos módulos: El primero está relacionado al proveedor de servicios y lo que se debe de tomar en cuenta para poder adquirirlo, ya que hoy en día existen muchos proveedores que cumplen con las expectativas de los clientes, pero no hay que dejar a un lado lo verdaderamente importante y es que hay que tomar en cuenta qué control lleva el proveedor de servicios para ofrecer soluciones de seguridad, así mismo qué herramientas tienen disponibles para cualquier tipo de situación que se pueda presentar. Y el segundo módulo describe la arquitectura a nivel de máquinas virtuales, contenedores y un nivel más detallado de cómo estará estructurada la división de educación a distancia y las herramientas que se deben de contemplar en este módulo para poder seguir la línea de seguridad que se quiere plantear para que en un futuro se pueda implementar.

#### **4.2.1. Nivel Externo - MSSP**

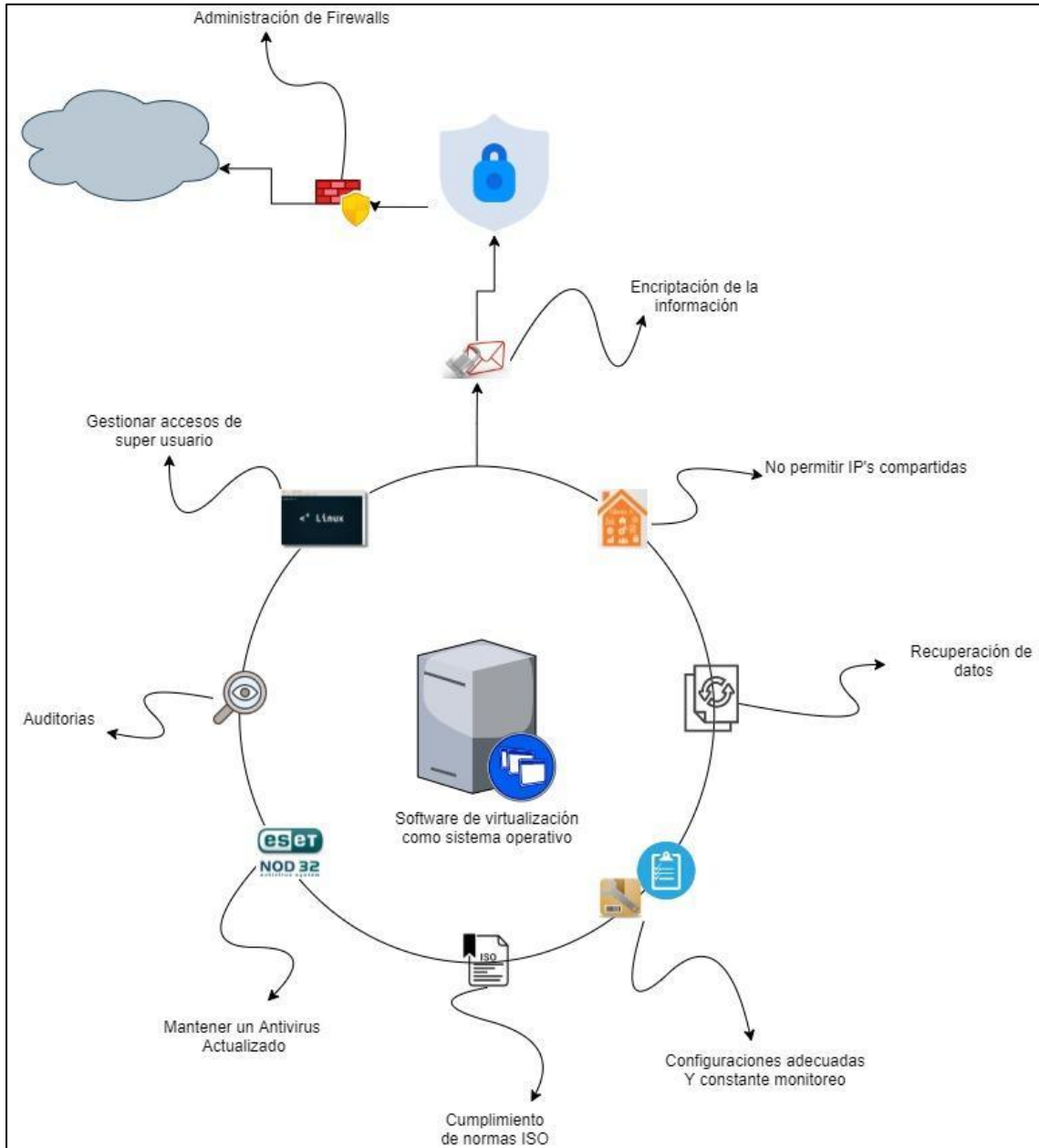
El proveedor de servicios de seguridad administrada por sus siglas en inglés MSSP "*Managed Security Service Provider*", este es un modelo que brinda monitoreo y administración de dispositivos o sistemas de seguridad, el cual tiene un amplio campo de herramientas para utilizar dentro de dicho modelo y para ellos es importante destacar que con este modelo los proveedores pueden adaptarse a las necesidades del cliente ya que ofrecen un paquete completo tanto de servicios y soluciones de TI y seguridad, esto a las empresas ya sean pequeñas o medianas parece prometedor ya que influye mucho en los costos que puedan tener dentro de su organización siendo esto

una opción para poder reducir no solo recursos sino menor tiempo de implementación, configuración y administración; así mismo menos personal, por lo que se vuelve una opción muy viable.

Algunos de los conceptos y herramientas que contempla el proveedor de servicios de seguridad administrada y que son importantes para este diseño de la arquitectura que abarcan casi todo lo que un proveedor debería de ofrecer ante cualquier momento situación son:

- Monitoreo
- Administración
- Administración de Firewalls
- Sistemas de detección de intrusos
- Servicios de red privada virtual (VPN)
- Auditorias
- Cumplimiento de normas
- Recuperación de datos
- Implementación de soluciones en tiempo real

Figura 8. **Arquitectura, parte 1**



Fuente: elaboración propia, empleando draw.io 2000.



Un tema importante que se debe considerar y que va implícito en el diseño de la arquitectura es la configuración, dentro de los MSSP es la administración y configuración que básicamente puede dejarse en manos de un proveedor de servicios que tendrá modelos y herramientas capaces para la gestión de incidentes de seguridad en tiempos menores a comparación si se hace por cuenta propia, ya que se puede correr el riesgo de que no exista alguna configuración adecuada y el constante monitoreo que llevará a anomalías en el servidor.

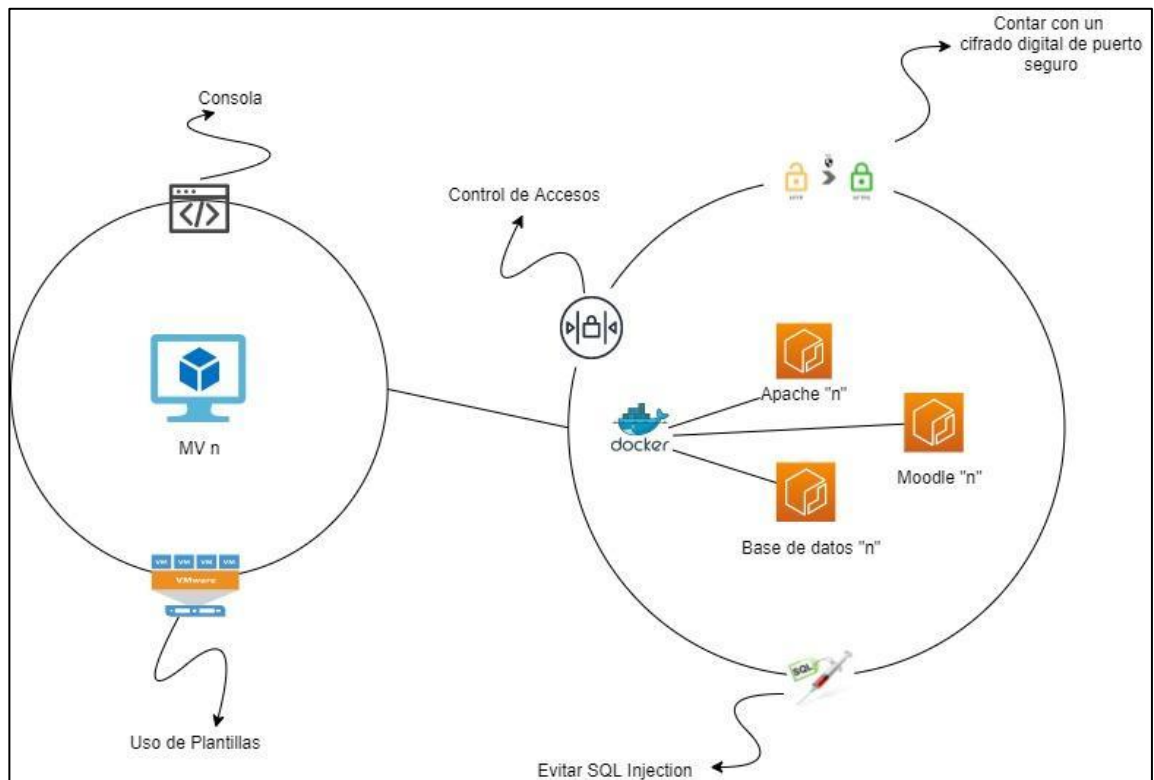
Respecto a las normas y políticas es importante considerar que el proveedor de servicios cuente con una certificación ISO, ya que esta certificación garantiza que la empresa se cuenta con altos estándares de seguridad. Algunas de las normas que se deben tener dentro de la lista serían: ISO 27001, 27701, 27017, 27018 y 27032 ya que estas son las que garantizan resguardar la integridad de la información de la empresa u organización. El objetivo de esta norma 27001 es que la empresa conozca los riesgos asociados al manejo de la información, con el fin de minimizarlos y gestionarlos por medio de un proceso eficiente y estructurado.

Con el tema de la recuperación de datos se recomienda utilizar un plan de recuperación ante desastres (DRP) por sus siglas en inglés (Disaster Recovery Plan) que esto se ha vuelto muy común que lo utilicen las empresas u organizaciones ante los ataques cibernéticos, fallos eléctricos u otra interrupción, por ende se han visto en la necesidad de utilizar uno, ya que entre mayor sea el tiempo de recuperación mayor será el impacto comercial, por lo tanto es de vital importancia contar con un *script* detallado de los pasos a seguir ante cualquier tipo de interrupción.

#### **4.2.2. Nivel Interno**

En esta sección se detalla la parte interna de la arquitectura ya que se cuenta con una máquina virtual que dentro de ella contiene un contenedor para administrar de forma estructurada la parte de los servicios a utilizar para la División De Educación a Distancia en Entornos Virtuales. En ella se detallan algunas prácticas que se pueden aplicar, por ejemplo si nos enfocamos en las páginas web como tal, que estarán expuestas al público (a estudiantes en este caso), se corre el riesgo de tener vulnerabilidades que pueden tener un nivel de riesgo alto, y es por ello que se recomienda contratar a un proveedor de servicios para el análisis de sus aplicaciones sometiéndolas a pruebas que harán que falle y que muestren todas las entradas a las que están expuestos estos servicios, así mismo se recomienda no colocar mayor detalle en los errores mostrados por las páginas, aunque esto pareciera una ayuda para el programador al momento que está desarrollando, con el hecho de colocar una aplicación en el IIS y dejar a vista de un atacante la versión del *framework* utilizada, ya que se le está abriendo una puerta a esa persona para que utilice cierto tipo de herramientas sofisticadas para lograr entrar y atacar, así mismo se recomienda tener todas las aplicaciones por puerto seguro 443 utilizando un cifrado digital; y así se puede enumerar todo tipo de herramientas que se pueden implementar para el diseño de la arquitectura.

Figura 9. **Arquitectura, parte 2**



Fuente: elaboración propia, empleando draw.io 2000.

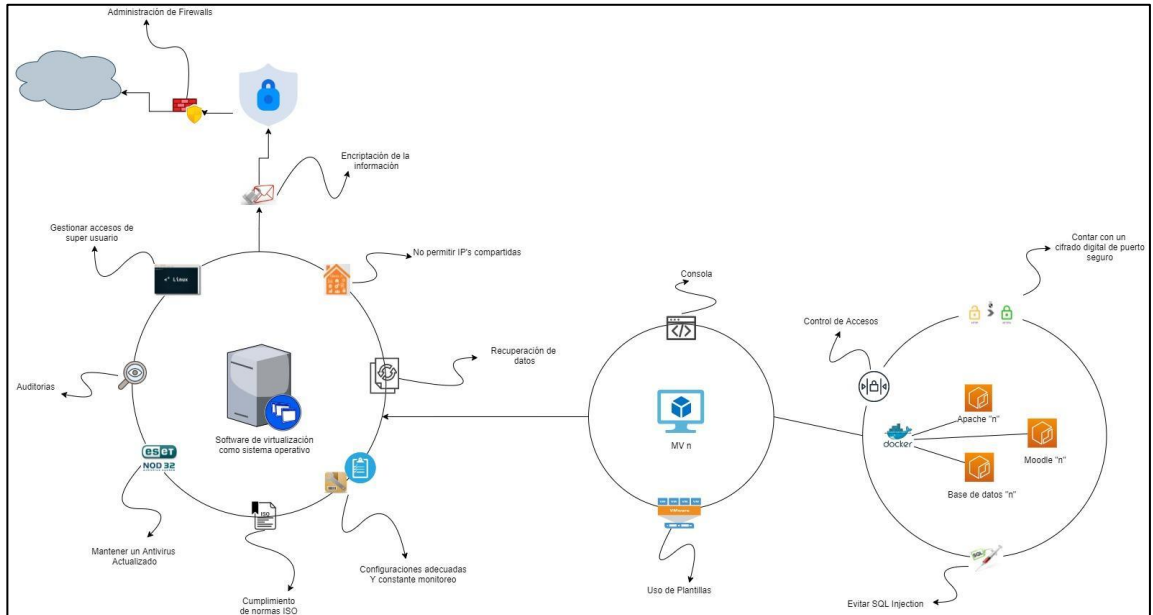
Así mismo se recomienda evitar ejecutar los contenedores en modo de superusuario, por ellos es importante contar con un control de accesos para limitar el acceso a la raíz del sistema operativo dejando este por defecto como modo lectura para el acceso, evitando así que los usuarios ingresen y realicen una mala configuración del contenedor dejando así un punto vulnerable, por lo tanto esta práctica puede ser muy efectiva para evitar tanto errores humanos como ataques de un ciberdelincuente.

Evitar SQL Injection, esto es muy común en temas de transacciones que están viajando de un lugar a otro ya que lo que se espera es poder alterar las

consultas que van directamente a la base de datos y poder filtrar o robar información que está viajando desde el servidor por ello es importante conocer este término y saber cómo protegerse del mismo para que nuestra información no caiga en manos equivocadas, se dará una breve explicación ya que para este tema se tiene mucha información y no es el fin de este capítulo desviarnos de lo verdaderamente importante. Pero, ¿cómo evitar esto? se detalla a continuación algunas de las medidas a considerar:

- Hacer uso de listas blancas: es importante contar con un *WhiteList* ya que sirve para indicar qué entidades son permitidas y bloquear todo lo que no se conoce, es mejor realizar esto a que se agregue a una lista negra ya que los atacantes pueden tener los conocimientos necesarios para eludir ser detectado por una lista negra, un claro ejemplo de cómo trabaja un *WhiteList* es el enfoque que se le da a un *firewall* ya que este permite solo determinadas direcciones IP para que puedan acceder a una red.
- Emplear mecanismos verificados: se debe buscar alguna tecnología moderna que ofrezca mecanismos de protección contra ataques SQLi y utilizar consultas parametrizadas o procedimientos almacenados.
- Mantener capacitado al equipo: esta es una parte crucial ya que los implicados en la creación de la aplicación deben conocer los riesgos a los que se enfrentan al poner su aplicación en producción y deben conocer sus vulnerabilidades, por ello es importante mantener capacitado a todo el personal o bien contratar a un proveedor que realice el análisis de vulnerabilidades a nivel más detallado.

Figura 10. Arquitectura propuesta



Fuente: elaboración propia, empleando draw.io 2000.



## CONCLUSIONES

1. La solución propuesta en este documento está enfocada en la utilización de una herramienta de virtualización llamada Proxmox, la cual se escogió como la ideal porque se adapta a los requerimientos solicitados por la División de Educación a Distancia en Entornos Virtuales, por su simplicidad y estructura; ya que permite la creación de máquinas virtuales que servirán como unidades académicas independientes y con una administración de recursos controlada que es esencial para la unidad en altas demandas.
2. El uso de una herramienta que sea eficiente al momento de la creación o implementación de aplicaciones puede reducir a gran escala los tiempos comparados a utilizar una herramienta convencional, es por ello que se utiliza Docker como herramienta para la administración de contenedores, ya que tiene la capacidad de estar en un entorno totalmente independiente y así utilizar los recursos de una manera eficiente adaptándolos a cualquier sistema, permitiendo así que este y muchos diseños más se puedan implementar de una forma fácil, rápida y automatizada.
3. La aplicación de buenas prácticas de seguridad lleva a la realización de una arquitectura robusta que se puede aplicar para la División de Educación a Distancia en Entornos Virtuales, ya que en ella se menciona lo realmente importante al momento de adquirir un servicio en la nube, así como algo esencial que es contar con un plan de

recuperación ante desastres, garantizar una certificación en normas ISO entre otras mencionada en el trabajo de investigación.



## RECOMENDACIONES

1. Solicitar, al proveedor de servicios, con quien se pueda contar, el uso de buenas prácticas de seguridad, que ayuden a mantener un sistema íntegro, confiable y disponible para la compañía, permitiendo así que los usuarios se sientan confiados de utilizar el servicio, sabiendo que están respaldados ante cualquier situación que pueda presentarse.
2. Utilizar herramientas que ayuden con el control de vulnerabilidades o bien optar por solicitar a un proveedor que pueda realizar el análisis de la o las aplicaciones que se tengan para mostrar al público, para que puedan revisar toda la funcionalidad, tanto de infraestructura como de aplicaciones web, esto con el fin de evitar que se puedan filtrar datos sensibles que utilicen en cada unidad académica.
3. Aprovechar las ventajas que tiene Docker para poder crear contenedores como plantillas para optar por las futuras migraciones hacia otro proveedor de servicios para evitar que se realice la configuración desde cero y lleve un trabajo extra, que es uno de los requerimientos solicitados por la DEDEV.
4. Mantener en constante monitoreo y actualización de todas las versiones de las herramientas que se vayan a utilizar en su futura implementación, ya que cuando un sistema está desactualizado puede ser un punto vulnerable para cualquier atacante; por lo tanto, se debe mantener la última versión de la herramienta Proxmox, Docker y las aplicaciones

que se vayan a instalar dentro de cada contenedor, esto con el fin de evitar tanto pérdidas de información ante algún fallo como un ataque.

## BIBLIOGRAFÍA

1. Check Point Software. (2021, July 21). MSSP - What is a Managed Security Service Provider? Extraído de <https://www.checkpoint.com/cyber-hub/network-security/what-is-a-managed-security-service-provider/>
2. Definición de seguridad informática. (n.d.). Extraído de <https://definicion.de/seguridad-informatica/>
3. Frachelle, A. (2018, November 21). 11 buenas prácticas de seguridad y protección de datos en empresas. Extraído de <https://www.idearius.com/es/blog/buenas-practicas-de-seguridad-y-proteccion-de-datos-en-empresas/>
4. Gutiérrez, P. (2017, June 21). ¿Qué es y como surge la criptografía?: Un repaso por su historia. Extraído de [https://www.genbeta.com/desarrollo/que-es-y-como-surge-la-criptografia-un-repaso-por-su-historia#:~:text=Los primeros mensajes cifrados que,forma longitudinal \(como en la](https://www.genbeta.com/desarrollo/que-es-y-como-surge-la-criptografia-un-repaso-por-su-historia#:~:text=Los primeros mensajes cifrados que,forma longitudinal (como en la)
5. MSSP Managed Security Service Provider Firewall Security. (n.d.). Extraído de <https://www.firemon.com/managed-security-service-providers/>
6. Proveedores de servicios de seguridad administrada: ¿qué son y qué ofrecen? (2017, February 01). Extraído de

<https://www.welivesecurity.com/la-es/2017/02/01/proveedores-de-servicios-de-seguridad-administrada/>

7. Rkempf. (n.d.). Prácticas recomendadas de seguridad para las máquinas virtuales. Extraído de <https://docs.vmware.com/es/VMware-vSphere/6.5/com.vmware.vsphere.security.doc/GUID-14CCC8CD-D90D-4227-B2C3-0A93D3C023BA.html>
8. Servidor proxy. (2021, July 01). Extraído de [https://es.wikipedia.org/wiki/Servidor\\_proxy](https://es.wikipedia.org/wiki/Servidor_proxy)
9. Virtualization Technology & Virtual Machine Software: What is Virtualization? (2021, August 04). Extraído de <https://www.vmware.com/latam/solutions/virtualization.html>
10. NEW: Proxmox VE 7.0. (n.d.). Extraído de <https://www.proxmox.com/en/>
11. ¿Qué es la virtualización? (n.d.). Extraído de <https://www.redhat.com/es/topics/virtualization/what-is-virtualization>
12. ¿Qué es un hipervisor? (n.d.). Extraído de <https://www.redhat.com/es/topics/virtualization/what-is-a-hypervisor>
13. ¿Qué es una máquina virtual? (n.d.). Extraído de <https://www.redhat.com/es/topics/virtualization/what-is-a-virtual-machine>

14. ¿Qué tan vulnerable es su proveedor de alojamiento web? (2021, April 29). Extraído de <https://www.webhostingsecretrevealed.net/es/blog/web-hosting-guides/how-vulnerable-is-your-web-hosting-provider/>



## APÉNDICE

Para realizar la implementación de la solución propuesta en los capítulos anteriores, se elaboró un manual para la “Configuración e implementación de contenedores Docker administrados, sistema de virtualización Proxmox para la aplicación en un entorno de educación en línea configurado por unidad académica”.

El manual se diseñó para realizar la configuración inicial del sistema operativo, en este caso en concreto, se utilizó la versión Debian Buster; las configuraciones de red del servidor, la instalación y configuración del software para el acceso remoto mediante el protocolo SSH, y *software* adicional para el sistema operativo, como paquetes para descargar el *software* necesario, búsqueda de ficheros, manejo de redes, editor de texto, por ejemplo; y por último, una comprobación de todas las configuraciones para validar que el sistema operativo está listo para la instalación de Proxmox, como el ‘hostname’ del servidor, el nombre de dominio, verificar las direcciones IP.

Para la instalación de Proxmox se detalla las verificaciones de los requerimientos para la virtualización de componentes del servidor, que el procesador del servidor soporte virtualización nativa; la configuración, descarga y actualización de paquetes de *software* necesarios para la instalación, como los repositorios y paquetes de Proxmox para Debian, servidor Samba, instalación y configuración del servidor de correo, el cargador de arranque ‘Grub’; y las configuraciones opcionales y requeridas para dar por finalizada la instalación de Proxmox VE, como la desinstalación del ‘OS prober’, actualizar el cargador de arranque ‘Grub’. Una vez finalizada la instalación, se detallan los

Continuación del apéndice 1.

pasos a seguir para comprobar que la instalación está correcta y configuraciones posteriores, como verificar que el sistema operativo se ejecuta utilizando el Kernel de Proxmox y no el de Debian, remover el kernel de Debian ya que no será de utilidad, verificar el servicio 'pveproxy', el acceso a la interfaz de Proxmox mediante el panel de autenticación, deshabilitar el acceso por el usuario 'root'.

El manual también contempla y detalla paso a paso la creación de máquinas virtuales a través de la interfaz gráfica de Proxmox VE. Se describen las configuraciones generales de la máquina virtual, como el nodo dónde se ejecutará la máquina virtual, el identificador único de la máquina virtual, el nombre de la máquina virtual, el conjunto lógico al que pertenece la máquina virtual, si la máquina debe arrancar cuando inicie el servidor anfitrión, el orden de inicio y apagado de la máquina virtual, el tiempo que debe esperar la máquina virtual para que se agregue a la cola de prioridad, el tiempo de espera antes de forzar el apagado de la máquina virtual; configuraciones del sistema operativo de la máquina virtual, como la imagen ISO, tipo de sistema operativo, versión del sistema operativo; configuración del sistema, como la tarjeta gráfica, el controlador SCSI, el 'Qemu Agent', BIOS, chipset; configuración de discos duros, como el tipo de Bus, número de Bus, unidad de almacenamiento, tamaño del disco, formato, emulación de SSD, límite de lectura y escritura; configuración del CPU, como la cantidad de sockets, la cantidad de núcleos, el tipo de procesador a virtualizar; la memoria RAM, como cantidad total de memoria, cantidad mínima de memoria, dispositivo 'Ballooning'; configuración de red, como controladores de interfaz de red, si no se desea habilitar ningún dispositivo de red, configurar el adaptador de red en modo 'Puente'; y el acceso y ejecución de la nueva máquina virtual desde la interfaz de Proxmox.



Continuación del apéndice 1.

La siguiente sección del manual detalla la instalación y configuración de la solución del software que utiliza la DEDEV para la plataforma educativa en línea, Moodle.

Para poder implementar las ventajas del uso de contenedores de *software*, el manual detalla la instalación y configuración de Docker. Dentro de la configuración se especifica la implementación de Moodle sobre contenedores de *software*.

Dentro del detalle de conceptos para aprovechar las características de Docker, se incluye una sección para crear plantillas de máquinas virtuales. Estas plantillas se pueden aprovechar para ahorrar recursos cuando se quiera implementar una nueva unidad académica. También cómo crear máquinas virtuales a partir de estas plantillas, cómo migrar las plantillas mencionadas, cuando se cambie de servidor por cualquier motivo.

Como anexos del manual se incluye la instalación de Debian Buster, en caso de que el servidor donde se desea instalar Proxmox no tenga un sistema operativo instalado. La conexión remota hacia el servidor mediante el protocolo SSH, utilizando una herramienta alterna con interfaz gráfica. Y la descarga de imágenes ISO que se utilizan para la creación de máquinas virtuales en Proxmox, mediante 'wget', o mediante el protocolo 'SCP'.

Por último, se detallan las sugerencias de seguridad para el servidor anfitrión, y mitigar problemas comunes a los que están expuestas estas soluciones.

Continuación del apéndice 1.

Para cada una de las secciones mencionadas, se detalla paso a paso para realizar cada configuración, cada paso detalla los comandos necesarios para llevarlo a cabo, y en los casos donde es necesario se incluyeron imágenes para que el usuario del manual pueda comprobar que está realizando una instalación correcta de Proxmox, Docker y Moodle.

Fuente: elaboración propia.