



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

**DOCUMENTACIÓN DE LA CREACIÓN DE ENTORNOS DE DESARROLLO
VIRTUALIZADOS Y REPLICABLES CON VAGRANT PARA LA PLATAFORMA E-
LEARNING MOODLE, PARA LA RED DE APOYO DIGITAL DOCENTE DE LA DIVISIÓN DE
EDUCACIÓN A DISTANCIA EN ENTORNOS VIRTUALES -DEDEV-, DE LA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Gustavo Adolfo Ichel Escobar

Kevin Oswaldo Mejía Lemus

Asesorados por el Ing. Hugo Esteban Ajuchan Chis

Guatemala, noviembre de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DOCUMENTACIÓN DE LA CREACIÓN DE ENTORNOS DE DESARROLLO
VIRTUALIZADOS Y REPLICABLES CON VAGRANT PARA LA PLATAFORMA E-
LEARNING MOODLE, PARA LA RED DE APOYO DIGITAL DOCENTE DE LA DIVISIÓN DE
EDUCACIÓN A DISTANCIA EN ENTORNOS VIRTUALES -DEDEV-, DE LA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

GUSTAVO ADOLFO ICHEL ESCOBAR

KEVIN OWALDO MEJÍA LEMUS

ASESORADOS POR EL ING. HUGO ESTEBAN AJUCHAN CHIS

AL CONFERÍRSELES EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE 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. Kevin Vladimir Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Cesar Augusto Fernández Cáceres
EXAMINADOR	Ing. Cesar Rolando Batz Saquimux
EXAMINADOR	Ing. Herman Igor Véliz Linares
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

Gustavo Adolfo Ichel Escobar

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. Kevin Vladimir Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADORA	Inga. Devora Emperatriz Meza Orellana
EXAMINADOR	Ing. Luis Fernando Espino Barrios
EXAMINADOR	Ing. Álvaro Giovanni Longo Morales
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

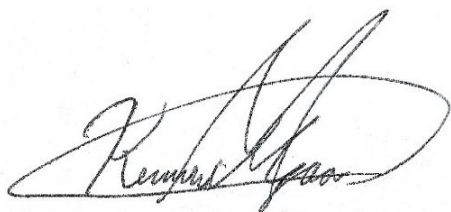
Kevin Oswaldo Mejía Lemus

HONORABLE TRIBUNAL EXAMINADOR

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

**DOCUMENTACIÓN DE LA CREACIÓN DE ENTORNOS DE DESARROLLO
VIRTUALIZADOS Y REPLICABLES CON VAGRANT PARA LA PLATAFORMA E-
LEARNING MOODLE, PARA LA RED DE APOYO DIGITAL DOCENTE DE LA DIVISIÓN DE
EDUCACIÓN A DISTANCIA EN ENTORNOS VIRTUALES -DEDEV-, DE LA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA.**

Tema que nos fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 1 de julio de 2020.



Kevin Oswaldo Mejía Lemus



Gustavo Adolfo Ichel Escobar



Ciudad de Guatemala, 20 de septiembre de 2021

Ing. Carlos Azurdia
Coordinador Trabajos de Tesis
Escuela de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería


Estimado Ing. Azurdia:

Por este medio informo que en mi papel de asesor he revisado el trabajo de Tesis titulado: **"DOCUMENTACIÓN DE LA CREACIÓN DE ENTORNOS DE DESARROLLO VIRTUALIZADOS Y REPLICABLES CON VAGRANT PARA LA PLATAFORMA E- LEARNING MOODLE, PARA LA RED DE APOYO DIGITAL DOCENTE DE LA DIVISIÓN DE EDUCACIÓN A DISTANCIA EN ENTORNOS VIRTUALES -DEDEV-, DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**, elaborado por los estudiantes **Gustavo Adolfo Ichel Escobar** quién se identifica con DPI **2368564130101** y registro académico **200312459**, así como **Kevin Oswaldo Mejía Lemus** quién se identifica con DPI **3032363530108** y registro académico **201612460**.

Con base a la evaluación realizada hago constar que este trabajo de graduación cuenta con mi aprobación.

Atentamente,

"Id y Enseñad a todos"


Ing. Hugo Esteban Ajuchan
Asesor de Tesis
Colégiado 9531



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

Guatemala, 30 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 **GUSTAVO ADOLFO ICHEL ESCOBAR** con carné **200312459** y CUI **2368 56413 0101** y **KEVIN OSWALDO MEJÍA LEMUS** con carné **201612460** y CUI **3032 36353 0108**, titulado: **“DOCUMENTACIÓN DE LA CREACIÓN DE ENTORNOS DE DESARROLLO VIRTUALIZADOS Y REPLICABLES CON VAGRANT PARA LA PLATAFORMA E-LEARNING MOODLE, PARA LA RED DE APOYO DIGITAL DOCENTE DE LA DIVISIÓN DE EDUCACIÓN A DISTANCIA EN ENTORNOS VIRTUALES -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 **“DOCUMENTACIÓN DE LA CREACIÓN DE ENTORNOS DE DESARROLLO VIRTUALIZADOS Y REPLICABLES CON VAGRANT PARA LA PLATAFORMA ELEARNING MOODLE, PARA LA RED DE APOYO DIGITAL DOCENTE DE LA DIVISIÓN DE EDUCACIÓN A DISTANCIA EN ENTORNOS VIRTUALES -DEDEV-, DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, realizado por los estudiantes, GUSTAVO ADOLFO ICHEL ESCOBAR y KEVIN OSWALDO MEJÍA LEMUS aprueba el presente trabajo y solicita la autorización del mismo.*

“IN V ENSEÑAN A TODOS”

Msc. Carlos Gustavo Alonzo
Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 09 de noviembre de 2021

DTG. 664.2021

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: **DOCUMENTACIÓN DE LA CREACIÓN DE ENTORNOS DE DESARROLLO VIRTUALIZADOS Y REPLICABLES CON VAGRANT PARA LA PLATAFORMA E-LEARNING MOODLE, PARA LA RED DE APOYO DIGITAL DOCENTE DE LA DIVISIÓN DE EDUCACIÓN A DISTANCIA EN ENTORNOS VIRTUALES -DEDEV-, DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por los estudiantes universitarios: **Gustavo Adolfo Ichel Escobar Kevin Oswaldo Mejía Lemus**, 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, noviembre de 2021

AACE/cc

ACTO QUE DEDICO A:

Dios	Porque toda buena dádiva viene de Él, y esta es una corona que con gozo entrego a su trono.
Mi esposa	Adriana Olmedo de Ichel. Por esforzarte junto conmigo para alcanzar esta meta.
Mi hijo	Benjamin Ichel. Por ser un motor que me impulsa día a día.
Mis padres	Pedro Ichel y Zoila Escobar de Ichel. Esta meta antes que mía es de ustedes siempre.
Mis hermanos	Willy, Ingrid y Carlos Ichel, cada uno me dio de sus dones, y les admiro.
Mis sobrinos	Para que esta estafeta la alcancen y la sobrepasen con creces.

Gustavo Adolfo Ichel Escobar

ACTO QUE DEDICO A:

Dios	Por darme la vida y permitir que el cauce natural de las cosas desembocara en este momento.
Mis padres	Oswaldo Mejía y Mirna Lemus, gracias por siempre brindarme su amor y apoyo a lo largo de mi vida.
Mis hermanos	Gracias por su cariño y apoyo constante.
Mi demás familia	Por siempre brindarme su apoyo y ser una influencia importante en mi vida.
Mi pareja	Mishell Cifuentes, gracias por brindarme su apoyo incondicional durante el desarrollo y realización de este proyecto.
Mis amigos	Emely García, Robin Salvatierra, Herbert Reyes y a todos mis amigos, por todos los momentos de alegría, tristeza y risa que hemos compartido. Gracias por su apoyo incondicional, sin duda no hubiera sido igual sin ustedes.

Kevin Oswaldo Mejía Lemus

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser el <i>alma máter</i> que empezó a pulir mi conocimiento.
Facultad de Ingeniería	Por enseñarme a sobrevivir académica y profesionalmente.
División de Educación a Distancia	Por permitir el desarrollo de nuestro proyecto de graduación.
Mis amigos de estudios	A los que aún lo son, y también a los que no, Jairo Luch, Alex Barahona, Billy Muñoz, Josué Polanco, Huber Flores, Jeny Martínez.
Compañeros de cierre	Celso Soto, Kevin Mejía y Fernando Hernández, por ser un impulso definitivo para graduarme.
Ing. Hugo Ajuchan	Por apoyarnos durante el desarrollo de nuestra investigación.

Gustavo Adolfo Ichel Escobar

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por abrirme las puertas y darme los recursos necesarios para poder crecer profesionalmente.
Facultad de Ingeniería	Por el conocimiento adquirido dentro de las instalaciones.
Mis amigos de la Facultad	Por ser una influencia importante en mi vida y acompañarme durante este proceso.
Ing. Hugo Ajuchan	Por su apoyo y colaboración en el desarrollo de este trabajo.

Kevin Oswaldo Mejía Lemus

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	III
GLOSARIO	V
RESUMEN.....	IX
OBJETIVOS.....	XI
INTRODUCCIÓN	XIII
1. ENTORNOS DE DESARROLLO VIRTUALIZADOS Y REPLICABLES.....	1
1.1. Máquina virtual	1
1.2. Hipervisor	2
1.3. ¿Qué es un entorno de desarrollo virtualizado y replicable?	3
1.4. Usos y beneficios de un entorno virtualizado y replicable	3
1.5. Opciones en el mercado.....	4
2. ENTORNOS DE DESARROLLO REPLICABLES CON VAGRANT	7
2.1. ¿Qué es y cómo funciona Vagrant?	7
2.2. ¿Para qué se utiliza?	8
2.3. Ventajas sobre el uso de contenedores	11
2.4. Requisitos para su instalación	12
2.5. Instalación de Vagrant.....	13
2.6. Comandos de uso	14
3. ANÁLISIS DEL PROCESO DE INSTALACIÓN DE MOODLE	17
3.1. Recopilación de información de la plataforma <i>e-learning</i> de la RADD.....	17

3.2.	Revisión de versiones	18
4.	ANÁLISIS DEL PROCESO DE AUTOMATIZACIÓN CON VAGRANT ...	23
4.1.	Configuración de Vagrant para la automatización de entornos replicables	23
4.2.	Automatización vía <i>script</i> de la instalación de prerequisites de Moodle	25
4.3.	Automatización vía <i>script</i> de la instalación de Moodle	27
4.4.	Unificación de las automatizaciones	30
5.	CONFIGURACIÓN, INSTALACIÓN Y AUTOMATIZACIÓN VÍA VAGRANT DE LA PLATAFORMA <i>E-LEARNING</i> MOODLE	33
5.1.	Instalación de prerequisites para Moodle y configuraciones iniciales	33
5.2.	Configuración de MYSQL y PHP	37
5.3.	Culminación del proceso de instalación	38
5.4.	Creación de entorno de Moodle replicable con Vagrant	39
CONCLUSIONES		47
RECOMENDACIONES		49
BIBLIOGRAFÍA		51

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Vagrant con Infraestructura AWS.....	10
2.	Instalación de Vagrant en Ubuntu	13
3.	Verificación de instalación de Vagrant	14
4.	Inicialización de un proyecto Vagrant.....	24
5.	Ejecución de un proyecto Vagrant	24
6.	Eliminación de un proyecto Vagrant.....	25
7.	<i>Script</i> de instalación de prerequisites Moodle.....	26
8.	Estructura de archivos de preparación para la instalación de Moodle	27
9.	Contenido del archivo time_zone.sql.....	27
10.	Contenido del archivo db_moodle.sql	28
11.	<i>Script</i> de instalación de Moodle	29
12.	Estructura de archivos del proyecto Vagrant y <i>scripts</i> de instalación de Moodle	30
13.	Configuración del proyecto Vagrant	31
14.	<i>Script</i> de instalación de prerequisites de Moodle	35
15.	Definición de zona horaria de MYSQL	36
16.	Configuración de parámetros de php.ini.....	36
17.	Copia de repositorio a ruta deseada	37
18.	Configuración de mysqld.conf	37
19.	Ejecución de sentencias en MYSQL	38
20.	Configuración de php.ini.....	38
21.	Ejecución de archivo de instalación install.php	38

22.	Configuración de permisos	39
23.	<i>Script</i> parametrizado de la instalación de Moodle y sus prerrequisitos	40
24.	Configuración de la base de datos.....	43
25.	Definición de zona horario de la base de datos	43
26.	Archivo de configuración de Vagrant	44
27.	Comando para iniciar la instalación	44

GLOSARIO

Comando	Instrucción proporcionada a un sistema informático desde una herramienta para línea de órdenes.
Contenedor	Paquete de elementos que permiten ejecutar una aplicación determinada en cualquier sistema operativo.
<i>Continuous Integration</i>	Proceso que se lleva a cabo dentro de servicios como plataforma en la nube, que permite integrar los cambios a los servicios involucrados de forma transparente.
<i>Continuous Deployment</i>	Proceso que va dentro de la integración continua, y que permite automatizar el proceso de despliegue de servicios en la nube como plataforma.
DEDEV	División de Educación a Distancia en Entornos Virtuales.
Docker	Tecnología que permite la creación de contenedores y uso de contenedores de Linux.
<i>E-learning</i>	Nuevo concepto de educación a distancia en el que se integra el uso de las tecnologías de la información y comunicación.

GNU	Por sus siglas en inglés: <i>General Public License</i> es una licencia de derecho de autor de software, que garantiza a los usuarios finales la libertad de usar, estudiarlo, compartirlo y modificarlo.
LMS	Por sus siglas en inglés: <i>Learning Management System</i> hace referencia a sistemas para la administración del aprendizaje.
LTS	Por sus siglas en inglés: <i>Long Term Support</i> , hace referencia a versiones de software con soportes durante un período más largo que el normal.
Manejador de base de datos	Software orientado al manejo de base de datos, cuya función es servir de interfaz entre la base de datos, el usuario y las distintas aplicaciones utilizadas.
Moodle	Es una herramienta de gestión de aprendizaje (LMS), concebida para ayudar a los docentes a crear comunidades de aprendizaje en línea.
Plataforma virtual	Espacios en Internet que permiten la ejecución de diversas aplicaciones o programas en un mismo lugar para satisfacer distintas necesidades.
RADD	Red de Apoyo Digital Docente.
Script	Secuencia de comandos que tienen como objetivo ejecutar distintas acciones o configuraciones.

Servidor Web	Software que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta.
Sistema operativo	Conjunto de órdenes y programas que controlan los procesos básicos de una computadora y permiten el funcionamiento de otros programas.
Software	Es un conjunto de programas capaz de realizar tareas lógicas y procesar información.
Ubuntu Server	Software específico para gestionar servidores, el cual solo tiene una interfaz de líneas de comando
Vagrant	Es una herramienta para la creación y configuración de entornos de desarrollo virtualizados.
VirtualBox	Es una aplicación para hacer máquinas virtuales con un sistema operativo.

RESUMEN

Por motivos de impulsar la educación a distancia por medio de las plataformas virtuales, se crea la División de Educación a Distancia en Entornos Virtuales de la Universidad de San Carlos de Guatemala, cuyo principal objetivo es el de fomentar la educación virtual a distancia y en línea. Dadas las circunstancias derivadas de la pandemia de coronavirus provocada por el virus SARS-CoV-2 en 2020, la división organizó y planificó ampliar el alcance de estas plataformas, de tal manera que estas fueron expuestas y ofrecidas a cada Facultad, Escuela, Centros Universitarios y Unidades Académicas de la Universidad.

La División de Educación a Distancia en Entornos Virtuales actualmente dispone de una plataforma virtual base, la cual puede ser utilizada por las distintas Unidades Académicas, Escuelas y Facultades que así lo requieran, así mismo la plataforma cuenta con las herramientas necesarias para realizar clases en línea por parte de los docentes de la Universidad de San Carlos de Guatemala, sin embargo, no cuenta con una base documentada del proceso de instalación, a fin de agregar una nueva unidad académica y pueda hacer uso de esta plataforma.

Además, el proceso de instalación si bien es sencillo, dada la gran cantidad de unidades con las que cuenta la Universidad se ve en la necesidad de automatizar este proceso, para replicar de una forma sencilla, automática y parametrizada la plataforma *e-learning* Moodle para incorporar nuevas unidades académicas que así lo requieran. Permitiendo así completar la incorporación de

una nueva unidad académica de manera más sencilla y rápida, además de disminuir el posible error humano al realizar una tarea repetitiva.

OBJETIVOS

General

Llenar los vacíos documentales con material escrito que permita generar un valor agregado en el proceso de transmisión de conocimiento relacionado con la instalación de la plataforma Moodle, para incorporar una unidad académica de forma automatizada a la red de apoyo digital docente por medio de una nueva instancia de la plataforma Moodle creada mediante entornos de desarrollo virtualizados y replicables con la herramienta Vagrant.

Específicos

1. Documentar el proceso de instalación y configuración, con la finalidad de ofrecer una guía que permita la replicación de los ambientes actualmente en producción y la puesta en marcha de la plataforma Moodle para las unidades académicas que tengan la necesidad de utilizarla.
2. Permitir que se incorpore de manera sencilla una nueva Unidad Académica a la Red de Apoyo Digital Docente de forma automatizada y parametrizada.
3. Flexibilizar el despliegue de instancias como máquinas virtuales en servicios por plataforma en la nube, sin la necesidad de apegarse a una marca en específico.

INTRODUCCIÓN

La División de Educación a Distancia en Entornos Virtuales fue creada para impulsar la educación a distancia, mediante el desarrollo de las modalidades de enseñanza-aprendizaje en línea. La cual tiene como objetivo los programas oficiales de la administración central y las distintas unidades académicas de la Universidad de San Carlos de Guatemala.

A través de la Red de Apoyo Digital Docente (RADD) se creó una plataforma virtual base que puede ser utilizada por las distintas unidades académicas, facultades y centros universitarios de la Universidad, esta plataforma permitirá llevar a cabo las actividades académicas por parte de los docentes de la Universidad de San Carlos de Guatemala de una forma adecuada y de manera virtual.

Derivado de las circunstancias provocadas por la pandemia del virus SARS-CoV-2 en 2020, surge este proyecto el cual tiene como objetivo implementar la automatización de la instalación de la plataforma *e-learning* Moodle a fin de agregar una nueva unidad académica de una forma sencilla, esto dado la gran cantidad de unidades que son parte de la Universidad, lo cual facilitará replicar la plataforma para cada unidad que lo solicite.

Este proceso de automatización busca realizar la instalación de la plataforma de manera parametrizada adecuándose así a las necesidades de cada unidad que lo necesite, con el objetivo que sea un proceso sencillo y transparente para los administradores de la plataforma. Para este proceso se llevó a cabo una serie de reuniones con el personal a cargo de administrar

dicha plataforma, las cuales permitieron conocer el proceso actual para su instalación y configuración. Este proceso debía ser documentado para facilitar la transmisión de conocimiento y realizar la automatización de este.

Posterior a las reuniones se desarrolló el proceso de automatización de la replicación de la plataforma *e-learning* Moodle y de forma posterior se creó el material escrito para la instalación, implementación y automatización, siendo esta última realizada con el uso de la herramienta Vagrant, la cual permite crear entornos virtuales de forma automática, logrando así replicar de una forma rápida y sencilla la plataforma y que pueda ser utilizada y configurada por cualquier unidad académica de la Universidad de San Carlos de Guatemala.

1. ENTORNOS DE DESARROLLO VIRTUALIZADOS Y REPLICABLES

Cuando se desarrolla una aplicación, plataforma o cualquier herramienta de software, es fundamental eliminar los errores relacionados entre las diferencias existentes entre un entorno de desarrollo y un entorno de producción. Así también cuando se cuenta con grupos grandes de desarrolladores, el hecho de contar con un entorno idéntico para cada integrante evitará problemáticas de compatibilidad, de configuraciones distintas, entre otros. Por esto, el uso de entornos virtualizados se vuelve una solución ante esta problemática y a continuación, se explicará su funcionamiento.

1.1. Máquina virtual

Una máquina virtual es un entorno que funciona como un sistema informático virtual, el cual cuenta con su propia CPU, memoria, interfaz de red y almacenamiento, la cual funciona mediante el uso de un software que es capaz de cargar un sistema operativo en su interior y hacerlo funcionar como lo haría un ordenador real.

Entonces, una máquina virtual se puede considerar como una emulación, la cual se encapsula en una serie de archivos. Estos archivos actúan como un contenedor desde el que se ejecuta la máquina virtual en una máquina física, con la gran ventaja de que nada de lo que suceda en el interior de la máquina virtual afecte al equipo que la ejecuta.

Al sistema de software que permite ejecutar máquinas virtuales se le conoce como hipervisor, este software se encarga de separar los recursos de la máquina física e implementarlos adecuadamente para que la máquina virtual sea capaz de utilizarlos.

1.2. Hipervisor

La tecnología de virtualización permite que una máquina física pueda compartir un sistema con varios entornos virtuales, esto se logra gracias a los hipervisores. Un hipervisor es un software encargado de gestionar el sistema de hardware y separar los recursos físicos de las máquinas virtuales. Estos recursos son divididos según las necesidades desde el entorno físico hacia las máquinas virtuales.

Además, el hipervisor se encarga de realizar la solicitud de recursos del sistema físico cuando una máquina virtual se encuentra en ejecución y demanda recursos adicionales, así la máquina virtual podrá acceder al grupo compartido de recursos físicos y completar sus actividades.

También es importante conocer que existen dos tipos de hipervisores:

- Hipervisor de tipo 1: se encuentra en equipos que no cuentan con un sistema operativo, por lo cual el hipervisor se encarga de programar los recursos de las máquinas virtuales directamente en el sistema de hardware.
- Hipervisor de tipo 2: a diferencia de los hipervisores de tipo 1, se encuentra alojado, por lo cual se encarga de programar los recursos de las máquinas virtuales en un sistema operativo *host* y este se encarga de ejecutarlo en el sistema de hardware.

1.3. ¿Qué es un entorno de desarrollo virtualizado y replicable?

Cuando se habla de un entorno de desarrollo virtualizado, básicamente se hace referencia al uso de una máquina virtual donde se encuentran instalados y configurados los distintos componentes necesarios para la ejecución y desarrollo de software, según sea requerido. Al agregar la replicabilidad, se refiere a que este entorno de desarrollo, con sus especificaciones concretas, puede ser fácilmente replicable y distribuido a cualquier número de personas y que cada una de ellas tendrá una instancia idéntica de este entorno y puede trabajar sobre este sin inconveniente.

Entonces, un entorno de desarrollo virtualizado y replicable permite tener un mismo ambiente de desarrollo a través de una máquina virtual, con una configuración y especificaciones concretas, que son fácilmente replicables y que pueden distribuirse de manera tal que todas las réplicas sean idénticas.

1.4. Usos y beneficios de un entorno virtualizado y replicable

Al momento de trabajar en proyectos de desarrollo de software es fundamental mantener la coherencia con los entornos de desarrollo, con el objetivo de eliminar errores relacionados con las diferencias entre los entornos de desarrollo que pueda tener cada desarrollador de software. Por lo que el uso de entornos de desarrollo virtualizados brinda los siguientes beneficios:

- Es posible montar un ambiente lo más similar posible al servidor que será utilizado en producción.
- El equipo de desarrollo comparte un mismo entorno virtual.
- Existen máquinas virtuales preconfiguradas para los diferentes virtualizadores que permiten ahorrar tiempo en descargas e instalación.

- Se puede borrar la máquina virtual en el momento que ya no es necesaria y ahorrar espacio.
- Es posible mantener el código en la maquina *host* y compartirlo con la máquina virtual, permitiendo así probarlo en diferentes ambientes.
- Ahorrar tiempo en el paso a producción gracias a que se evitan errores por entornos de ejecución distintos.

Gracias a estos beneficios, los entornos de desarrollo virtualizados y replicables son una solución que permite trabajar sobre un ambiente de ejecución estándar, con especificaciones concretas, el cual es fácilmente replicable. Permitiendo así desarrollar distintas soluciones de software y evitar situaciones problemáticas en su paso a producción, tales como un funcionamiento incorrecto o problemas de compatibilidad, causados por entornos de ejecución distintos.

1.5. Opciones en el mercado

Herramientas para virtualización existen varias, pero dada la necesidad de que sean entornos virtualizados replicables las opciones disponibles que más se adecuan al objetivo del proyecto son las siguientes:

- Terraform: es una tecnología que permite describir un conjunto complejo de infraestructura tanto de forma local como remota. Esta herramienta se centra en construir y cambiar la infraestructura a lo largo del tiempo. Además, permite reproducir los aspectos mínimos del ciclo de vida de una máquina virtual, por lo cual a veces genera confusión con la herramienta Vagrant.

- Vagrant: es una herramienta que cuenta con una serie de funciones de nivel superior en el uso de máquinas virtuales, tales como: carpetas sincronizadas, redes automáticas, tunelización HTTP y muchas más características, las cuales permiten facilitar el uso de entornos de desarrollo. Esta es una tecnología que se centra en los entornos de desarrollo que, a comparación de Terraform, están fuera de su alcance.
- Docker: es una herramienta que permite la creación de contenedores ligeros y portables, que ejecutan aplicaciones de software en cualquier máquina con Docker instalado, sin depender de un sistema operativo. En los contenedores se empaqueta la información necesaria para ejecutar las soluciones de software. Además, estos contenedores son fácilmente replicables y estandarizados, permitiendo también facilitar los despliegues.

En base a lo descrito anteriormente, la herramienta que más se adecua a las necesidades de la RADD para automatizar el proceso de instalación de la plataforma *e-learning* Moodle es Vagrant, esto debido a que se enfoca directamente en la automatización de entornos de desarrollo, donde como se verá en capítulos posteriores, con esta herramienta se facilitará la incorporación de una nueva unidad académica de forma automatizada, parametrizada y sencilla. Además, permitirá agregar nuevas funciones a la plataforma, realizar pruebas y tener un paso a producción de forma fácil y sin inconvenientes.

2. ENTORNOS DE DESARROLLO REPLICABLES CON VAGRANT

2.1. ¿Qué es y cómo funciona Vagrant?

Hoy en día existe la necesidad, ya sea por motivos provenientes de la pandemia Covid-Sars-19 o como medio de la desconcentración del recurso humano dentro de una organización, de que el personal se encuentre en distintas ubicaciones a nivel global, esto no significa que hasta antes de 2020 esta disposición no existiese o no se hiciese, sino más bien por la necesidad de disminuir el contacto humano, la medida ha crecido a escala mundial, y esto es mucho más tangible en la industria de la informática.

Las empresas que basan sus ingresos o algunos de sus ingresos en que provengan de esta área, han tomado la decisión de que los recursos puedan trabajar ya sea en Guatemala, El Salvador, Estados Unidos, Brasil, o incluso desde Europa, en países como España u Holanda.

Debido a esta descentralización del capital humano, o más específico de la descentralización de los desarrolladores, arquitectos o aseguradores de calidad, surge la necesidad de establecer entornos de desarrollo lo más rápida y parecidos a los ambientes de calidad, UAD o producción con el objetivo de que la rápida adaptación de integrantes de un equipo.

La integración de cambios a nivel de código o configuración, e inclusive el despliegue centralizado de un proyecto de informática.

Existiendo herramientas que facilitan dichas tareas, tal cual las de *continuous integration*, *continuous deployment* o los gestores de versionamiento, también existen aquellas que permiten la generación de entornos de desarrollo muy famosos tales como Docker, utilizando virtualización a nivel de sistema operativo.

Sin embargo, también una opción que ha cobrado fuerza en los últimos años es Vagrant, que permite la creación de archivos de configuración vía *scripts* para generar un ordenador que trabaje mediante software utilizando una capa diferente a nivel de sistema operativo, más conocidos como máquinas virtuales, empleando al igual que GIT, virtualización a nivel de sistema operativo o mejor conocido como Hipervisor del tipo 2.

2.2. ¿Para qué se utiliza?

En los últimos ocho años el uso de servicios en la nube ha crecido de manera exponencial, y dentro de la gama que existe de tipos de servicios están los de plataforma, servicios que permiten la utilización de arquitecturas definidas por los fabricantes de la marca o casa que uno elija, estos permiten el uso de instancias de máquinas virtuales, serverless, funciones en la nube, gateways, corta fuegos, balanceadores de carga, y otros.

Es acá donde Vagrant entra en acción, la definición del fabricante como un software que permite la creación de entornos de desarrollo no es una de las únicas bondades que este ofrece.

Existe otra parte de la definición que da el fabricante y que integra dos conceptos que ya tocamos con anterioridad, la integración y el despliegue continuos, permitiendo así que la creación de una máquina virtual utilizando el

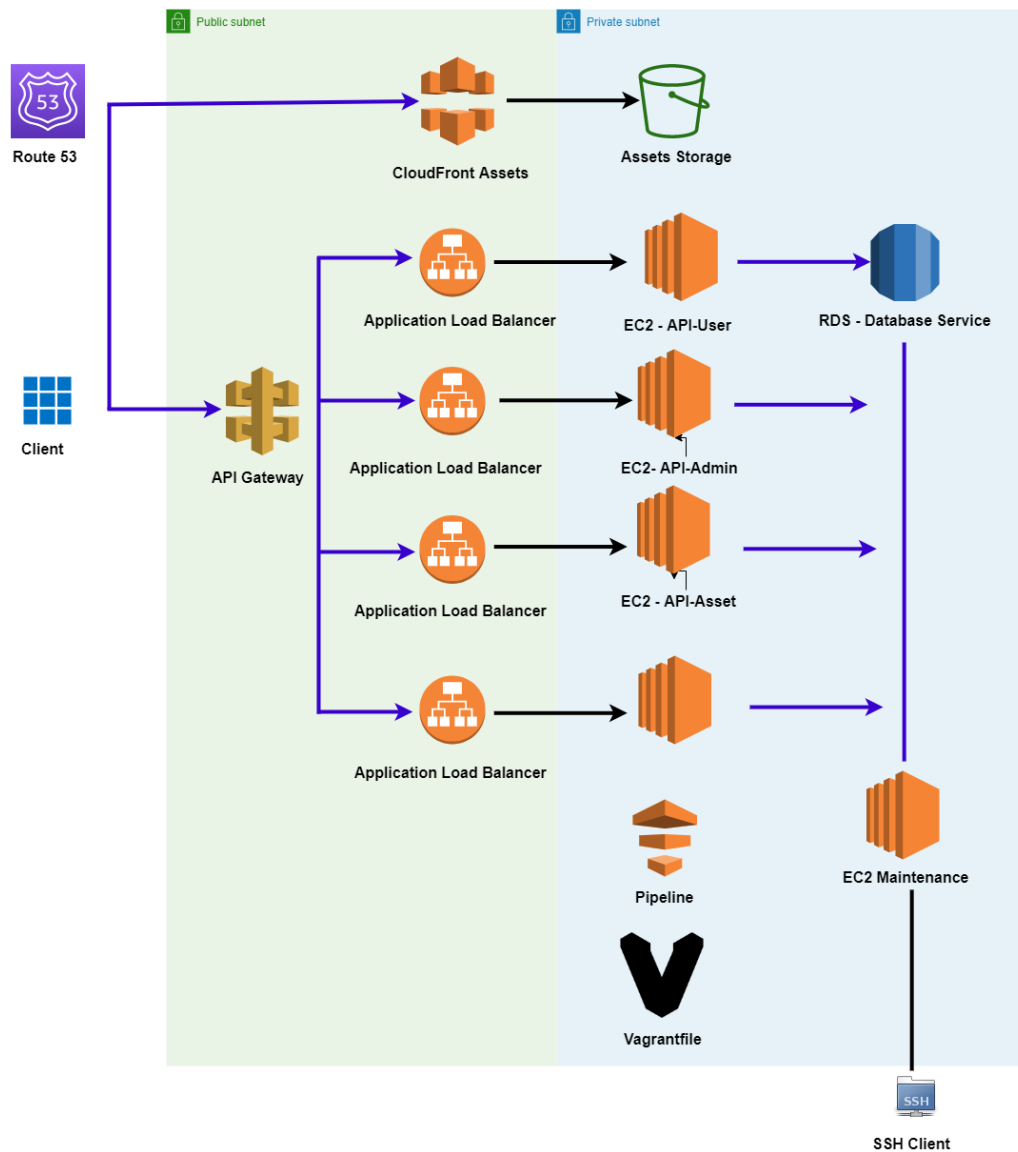
servicio en la nube de instancias de máquinas virtuales se integren, y la integración continua permita que esta instancia entre o case dentro de una arquitectura definida según la necesidad del proyecto, o de la plataforma que vaya a utilizarlas.

Pero lo descrito anteriormente remarca en términos conceptuales las bondades de Vagrant. A continuación, se listará el contexto de un ejemplo real de cómo puede ser utilizado.

- Definición de la arquitectura utilizando instancias de máquinas virtuales, pueden ser servicios como EC2-AWS, Instancias-Google Cloud, y otros.
- Integración de estos servicios con otros como base de datos, workers, Api's de terceros.
- Uso de la instancia para ofertar *backend* o *frontend*.
- Uso de la instancia para capa de persistencia con uso de base de datos relacional o no relacional.
- Generación de máquina virtual para la instancia, como entorno de desarrollo del equipo, utilizando Vagrant.
- Despliegue continuo a la instancia en la nube utilizando *script* de creación de entorno de desarrollo.
- Integración continua de la instancia creada desde Vagrant con la arquitectura de toda la solución.

Figura 1. **Vagrant con Infraestructura AWS**

Vagrant with AWS Infrastructure



Fuente: elaboración propia, empleando Google Graw IO.

2.3. Ventajas sobre el uso de contenedores

Es muy común que dentro del uso de los servicios como plataforma que proveen Google Cloud, AWS, Digital Ocean, y otros. La interrogante principal es si se utilizan servidores o los denominados serverless; dentro de la arquitectura, existen pros y contras en cada uno de estos servicios, y estos pueden ser costos, liviandad, escalabilidad o capacidad de respuesta entre otros.

Sin embargo, el uso de Vagrant como medida de creación de entorno viene denotado en cuál es el objetivo de la arquitectura y la posible envergadura que se necesite para tomar la decisión entre un servidor o un contenedor, por lo que se lista a continuación esas ventajas.

- Permite el uso de una alta envergadura de recursos como memoria, procesador al virtualizar directamente una instancia de máquina virtual, aunque su costo de una instancia es directamente proporcional a la configuración de esta.
- En entornos de desarrollo basados en Windows, no necesita de utilizar el hipervisor por defecto de Microsoft (Hyper-V), por lo que no es necesario utilizar la versión profesional.
- El uso de Docker como herramienta consume muchos más recursos que las máquinas virtuales que Vagrant puede utilizar, tal es el caso de VirtualBox, KVM, Xen o VMware.

2.4. Requisitos para su instalación

De acuerdo con el sitio oficial de Vagrant, los requerimientos mínimos de instalación son los siguientes para la versión 1.3.5:

- CPU 2.0 GHz
- Memoria RAM 4 GB
- OS: Versiones recientes de Linux basadas en Debian, Fedora o Red Hat
- OS: Windows 7 o posterior
- Tarjeta de video de 1GB de RAM o mayor
- 1200 MB de espacio en disco

Vale la pena resaltar que estos requerimientos atribuyen gran parte de los recursos a lo que necesitará el Hipervisor que Vagrant utilizará por medio de su línea de comandos, y el resultado óptimo del desenvolvimiento de la máquina virtual dependerá mucho también de los requerimientos mínimos que los hipervisores necesiten. Estos pueden ser verificados en los sitios oficiales de cada uno, a continuación, se listan los que Vagrant utiliza:

- Virtual Box (proveedor por defecto).
- VMware.
- Docker.
- Hiper-V.
- Proveedores personalizados: son hipervisores que determinada marca ha configurado para que pueda ser utilizado su producto en específico, utilizando un *plugin* oficial de Vagrant.
 - AWS
 - Google Cloud
 - Docker

2.5. Instalación de Vagrant

Ya que Vagrant es una herramienta que permite administrar máquinas virtuales, capaz de automatizar procesos, ahorrando tiempo de desarrollo, ejecución y elaboración de ambientes. Es una herramienta de gran valor que será utilizada en este proyecto.

Como se verá en capítulos posteriores, la red de apoyo digital docente cuenta con una plataforma *e-learning* Moodle que se encuentra instalada en un servidor con sistema operativo Ubuntu Server 18.04 LTS. Por lo tanto, el proceso de instalación estará enfocado para este sistema, sin embargo, es posible consultar su página oficial para conocer el proceso de instalación para cualquier otro sistema: <http://www.vagrantup.com/>.

En base a lo anterior, para el proceso de instalación en un servidor con Ubuntu Server 18.04 LTS es necesario ejecutar los comandos que se definen en la figura 2, en el servidor.

Figura 2. Instalación de Vagrant en Ubuntu

```
$ curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -  
$ sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb_release -cs) main"  
$ sudo apt-get update && sudo apt-get install vagrant
```

Fuente: elaboración propia, empleando VS Code.

Al finalizar la ejecución de los comandos anteriores, la instalación habrá concluido.

Es posible verificar si la instalación fue exitosa utilizando el comando de la figura 3. Si todo está correcto, se mostrarán comandos y opciones de uso.

Figura 3. **Verificación de instalación de Vagrant**

```
$ vagrant
Usage: vagrant [options] <command> [<args>]

    -v, --version          Print the version and exit.
    -h, --help             Print this help.

# ...
```

Fuente: elaboración propia, empleando VS Code.

En este punto es requisito instalar un software de virtualización. Para esto se recomienda instalar Virtual Box para preparar la inicialización y ejecución de un proyecto en Vagrant.

2.6. **Comandos de uso**

Vagrant posee una línea de comandos conocida como Vagrant CLI (*Command Line Interface*), la cual permite la interacción desde una terminal a modo de configurar o utilizar la interfaz de acuerdo con las necesidades requeridas. A continuación, se listan los comandos más utilizados y su respectiva descripción, sin embargo, para ver la lista de parámetros y sufijos que cada uno tiene se pueden ver en la guía oficial de Vagrant:

- Vagrant up: permite la creación y el lanzamiento de una máquina virtual junto con su configuración, definida dentro del archivo Vagrantfile.
- Vagrant destroy: permite detener la ejecución de una máquina virtual creada y gestionada desde Vagrant, así como la destrucción de todos los recursos que esta haya tenido configurada.

- Vagrant init: permite la inicialización de un entorno de desarrollo, creando un archivo de configuración Vagrantfile definiendo los valores predeterminados que necesita una máquina virtual.
- Vagrant upload: comando que realiza la carga de medios o archivos desde el *host* al huésped.
- Vagrant ssh: por medio de protocolo ssh, permite acceder a una terminal desde el equipo huésped.

3. ANÁLISIS DEL PROCESO DE INSTALACIÓN DE MOODLE

La red de apoyo digital docente (RADD) actualmente cuenta con una plataforma *e-learning* base de Moodle que ha sido personalizada de acuerdo con las necesidades de la Universidad. Esta plataforma permitirá dar continuidad a los servicios educativos que provee la Universidad de San Carlos de Guatemala ante cualquier cierre de las instalaciones físicas, provocada por factores que pueden poner en riesgo la integridad de la comunidad universitaria, como es el caso de la pandemia del coronavirus.

Debido a la existencia del uso de esta plataforma en la Universidad, es necesario realizar un proceso de análisis que se centre en conocer el ambiente que utiliza la RADD actualmente para el manejo de esta herramienta, tomando nota de los requisitos de instalación y el proceso necesario para su instalación y configuración.

3.1. Recopilación de información de la plataforma *e-learning* de la RADD

Partiendo de que la red de apoyo digital docente cuenta con un ambiente productivo para el manejo de la plataforma *e-learning* Moodle, con el objetivo de conocer este ambiente y el proceso de instalación requerido para incorporar una nueva unidad académica se llevó a cabo una serie de reuniones con el personal a cargo de administrar las distintas instancias de esta plataforma de la RADD.

Gracias a esta serie de reuniones se comprendió el proceso requerido para la creación de una nueva instancia de Moodle, su configuración, el sistema operativo utilizado y la versión de Moodle empleada. Con este conocimiento y con el propósito de tener un panorama más amplio sobre los componentes necesarios para la ejecución de la herramienta se realizaron instalaciones de prueba en distintos sistemas operativos, complementando así el conocimiento transmitido por el personal a cargo de administrar la plataforma Moodle. Estas instalaciones de prueba fueron realizadas en los siguientes sistemas operativos y contenedor de software:

- Ubuntu Linux 18.04 LTS
- Microsoft Windows 10 Pro
- Docker

Durante estas instalaciones de prueba, se tomó nota de los factores más importantes que pueden influir en la instalación de Moodle, para facilitar así la automatización de este proceso.

3.2. Revisión de versiones

Moodle es una plataforma de aprendizaje enfocada en proporcionar a educadores, administradores y estudiantes un sistema integrado único, robusto y seguro, es una herramienta que permite crear ambientes de aprendizaje personalizables. Esto permitió a la Unidad de Red de Apoyo Digital Docente (RADD) crear su propia plataforma base personalizada de Moodle, permitiendo de esta forma beneficiarse del costo/beneficio, flexibilidad y otras ventajas de las cuales provee esta herramienta hacia las distintas unidades académicas, facultades y centros universitarios de la Universidad de San Carlos de Guatemala.

A continuación, se lista los prerequisites principales de Moodle, los cuales son independientes del sistema operativo donde se instale:

- Sistema operativo: puede ser cualquier sistema que corra el siguiente software, tomando en cuenta el desempeño que se necesite y las habilidades que se tienen disponibles. Las elecciones más comunes son Linux y Windows, siendo Linux el sistema operativo que se considera óptimo para el manejo de esta plataforma.
- PHP: se requiere un conjunto de extensiones propias de PHP que sirven para la interpretación del código de programación bajo este lenguaje, las cuales permitirán completar la instalación de Moodle sin ningún inconveniente.
- Manejador de bases de datos: los manejadores de base de datos más comúnmente utilizados para esta plataforma son MySQL, MariaDB y PostgreSQL. Se recomienda utilizar cualquiera de estos manejadores de base de datos, además de contar con la ventaja de tener una abundante documentación y soporte. Y dependiendo de la base de datos por utilizar se deberá utilizar la extensión de PHP apropiada.
- Servidor web: los principales servidores web utilizados son Apache o IIS. El servidor que se elija debe estar configurado para servir adecuadamente archivos PHP. También, aunque la versión utilizada del servidor web no es crítica, es recomendable instalar la versión más reciente que se encuentre disponible.

Actualmente la RADD para la administración de la plataforma Moodle cuenta con el sistema operativo Ubuntu Server 18.04 LTS, al tratarse de un

Sistema Operativo Linux con Kernel GNU, los prerequisites podrían necesitar de instalación de paquetes de software o la actualización de estos.

Conociendo los prerequisites necesarios para la instalación de Moodle y sabiendo que, al hacer uso de una plataforma como esta, la cual necesita otros prerequisites para funcionar, es muy importante identificar las versiones adecuadas de cada prerequisite para permitir una correcta ejecución de la herramienta.

A continuación, se describen las versiones tanto de Moodle como de los prerequisites que se utilizarán en el proceso instalación:

- Moodle en su versión 3.8.2 (Build: 20200309) con modificaciones respecto a cualquier versión estándar desarrolladas por la Red de Apoyo Digital Docente.
- PHP versión 7.2 y sus extensiones.
 - php7.2-curl
 - php7.2-zip
 - php7.2-gd
 - php7.2-intl
 - php7.2-xmlrpc
 - php7.2-soap
- Librerías necesarias para Moodle en Ubuntu Server 18.04 LTS.
 - *ghostscript*
 - libgs-dev
 - gs-esp

- MySQL 8.0.20
- Apache 2.0

Una vez se ha identificado las versiones de cada uno de los prerequisites y de Moodle, es posible proceder con el siguiente paso, la instalación en conjunto, la cual se llevará a cabo generalmente en el Sistema Operativo Ubuntu Server 18.04 LTS, siendo este el sistema operativo con el actualmente cuenta la RADD.

Pero antes de iniciar con la instalación es necesario hacer una actualización del repositorio de paquetes de software que el sistema operativo utiliza, esta actualización puede realizarse utilizando el gestor de paquetes que viene por defecto, esto debido a que este gestor lleva un control del árbol de dependencias y de las versiones necesarias para no crear conflictos entre paquetes.

Luego de realizar actualización del árbol de dependencias y de los paquetes necesarios, se puede iniciar la instalación de los prerequisites, para dar paso a la instalación de la plataforma base personalizada de Moodle de la RADD, el desarrollo de este proceso se explicará en el capítulo 5.

4. ANÁLISIS DEL PROCESO DE AUTOMATIZACIÓN CON VAGRANT

Anteriormente se describió que es Vagrant su uso y beneficios, pero a manera de resumen Vagrant es una herramienta que permite administrar máquinas virtuales de una manera fácil, permitiendo automatizar procesos para ahorrar tiempo de desarrollo, ejecución y elaboración de ambientes. Provee un uso configurable, replicable y estable para los ambientes de trabajo utilizando un flujo que permite mayor productividad y flexibilidad en el desarrollo de un proyecto.

Vagrant permitirá a la Red de Apoyo Digital Docente contar con un entorno virtualizado y replicable el cual facilitará la incorporación de una nueva unidad académica de la Universidad de San Carlos de Guatemala para tener acceso al uso de esta plataforma para fines académicos. Esto le permite también ofrecer mejores tiempos en los procesos de instalación y configuración, así como un mejor control de las instancias de la plataforma que existan en producción.

4.1. Configuración de Vagrant para la automatización de entornos replicables

Para realizar la automatización de un entorno replicable es necesario contar con un software de virtualización instalado. Es recomendable instalar VirtualBox para preparar la inicialización y ejecución del proyecto.

Para la configuración de Vagrant primero es necesario crear un proyecto, para ello en la terminal del sistema operativo, en este caso Ubuntu Server 18.04

LTS se utiliza el comando de la figura 4. Este, a su vez, creará un archivo llamado Vagrantfile.

Figura 4. Inicialización de un proyecto Vagrant

```
$ vagrant init hashicorp/bionic64
```

Fuente: Vagrant – Init. *Proyecto Vagrant*. www.vagrantup.com/docs/cli/init. Consulta: 15 de agosto de 2021.

Al ejecutar este comando el proyecto quedará inicializado con una configuración predeterminada por Vagrant. El siguiente paso es probar la ejecución con esta configuración y verificar que todo sea ejecutado correctamente. El comando para ejecutar el proyecto se muestra en la figura 5.

Figura 5. Ejecución de un proyecto Vagrant

```
$ vagrant up
```

Fuente: Vagrant – Up. *Proyecto Vagrant*. www.vagrantup.com/docs/cli/up. Consulta: 15 de agosto de 2021.

Si todo inicia correctamente, seguidamente se prefiere eliminar este ambiente creado por la prueba anterior con el comando de la figura 6. Esta prueba simplemente es para verificar la correcta ejecución de la herramienta y que los requisitos estén instalados correctamente.

Figura 6. **Eliminación de un proyecto Vagrant**

```
$ vagrant destroy
```

Fuente: Vagrant. *Destroy*. www.vagrantup.com/docs/cli/destroy. Consulta: 15 de agosto de 2021.

4.2. Automatización vía *script* de la instalación de prerequisites de Moodle

Los prerequisites necesarios para la instalación de Moodle se listaron en el Capítulo 3 con más detalle, pero a manera muy general los prerequisites necesarios son los siguientes:

- PHP 7.2 y sus extensiones
- MySQL 8.0.20
- Apache 2.0

El *script* de instalación de estos prerequisites se define en un archivo llamado moodle.sh. Este *script* contiene la actualización de repositorios del sistema operativo, instalación del manejador de base de datos MySQL, instalación de PHP y sus extensiones, así como las extensiones necesarias para el funcionamiento de Moodle. Por último, se realiza la instalación del servidor web, que en este caso será Apache.

El *script* utilizado se muestra en la figura 7.

Figura 7. **Script de instalación de prerequisites Moodle**

```
#!/bin/bash
sudo apt-get update

#instalación de php 7.2
sudo apt-get install -y php7.2

##instalación de mysql
sudo apt-get install -y mysql-server mysql-client

#Definir hora local en el servidor
date
#timedatectl list-timezones | grep -i america
sudo unlink /etc/localtime
sudo ln -s /usr/share/zoneinfo/America/Guatemala /etc/localtime
date

#Librerías necesarias de Moodle
sudo apt-get install -y php7.2-curl
sudo apt-get install -y php7.2-zip
sudo apt-get install -y php7.2-gd
sudo apt-get install -y php7.2-intl
sudo apt-get install -y php7.2-mysql
sudo apt-get install -y php7.2-ldap
sudo apt-get install -y php7.2-mbstring
sudo apt-get install -y php7.2-xmllrpc
sudo apt-get install -y php7.2-xml
sudo apt-get install -y php7.2-soap
sudo apt-get install -y ghostscript
sudo apt-get install -y libgs-dev
sudo apt-get install -y gs-esp

#instalación de apache
sudo apt-get install -y apache2
```

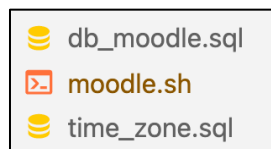
Fuente: elaboración propia, empleando VS Code.

4.3. Automatización vía *script* de la instalación de Moodle

Para facilitar la instalación personalizada de Moodle, los *scripts* se dividen en varios archivos. Estos archivos se encuentran estructurados como se muestra en la figura 8, a continuación, se describe el contenido de cada archivo.

- moodle.sh: contiene el *script* de instalación de los prerequisites y de Moodle personalizado.
- time_zone.sql: contiene el *script* para definir la zona horaria de la base de datos utilizada para la plataforma de Moodle.
- db_moodle.sql: contiene el *script* de la estructura de la base de datos de para la plataforma de Moodle.

Figura 8. **Estructura de archivos de preparación para la instalación de Moodle**



Fuente: elaboración propia.

El archivo time_zone.sql simplemente define la zona horaria de Guatemala para la base de datos. Su contenido está en la figura 9.

Figura 9. **Contenido del archivo time_zone.sql**

```
SET time_zone = '-6:00';
```

Fuente: elaboración propia, empleando VS Code.

El archivo db_moodle.sql contiene el *script* que define la estructura de la base de datos de Moodle, su codificación, sus credenciales y los permisos de acceso. La figura 10 define el *script* de este archivo.

Figura 10. **Contenido del archivo db_moodle.sql**

```
CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

CREATE USER 'moodledude'@'localhost'
IDENTIFIED BY 'passwordformoodledude';

GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,CREATE TEMPORARY
TABLES,DROP,INDEX,ALTER ON moodle.*
TO 'moodledude'@'localhost'
IDENTIFIED BY 'passwordformoodledude';
```

Fuente: elaboración propia, empleando VS Code.

Utilizando el mismo archivo moodle.sh se adjunta el *script* de la figura 11, que contiene los comandos para obtener los archivos de instalación, los permisos de directorios, la configuración horaria, la configuración del servidor web con PHP, la instalación y configuración de la base de datos con MySQL y la instalación de Moodle con la información necesaria.

Figura 11. **Script de instalación de Moodle**

```
#clonar repositorio de moodle
sudo git clone -b MOODLE_38_STABLE git://git.moodle.org/moodle.git /var/www/html/moodle

sudo mkdir /var/moodledata

#permisos de directorios
sudo chown -R www-data /var/moodledata
sudo chmod -R 777 /var/moodledata
sudo chmod -R 0755 /var/www/html/moodle

#configurar mysql para UTC -6:00
sudo mysql < /vagrant/time_zone.sql

#configurar servidor php
sudo sed -i -e '$upload_max_filesize = 1M' /etc/php/7.2/apache2/php.ini
sudo sed -i -e '$post_max_size = 1M' /etc/php/7.2/apache2/php.ini

sudo sed -i -e '$extension = php_mysqli' /etc/php/7.2/apache2/php.ini

#configurar mysql
sudo sed -i -e '$default_storage_engine = innodb' /etc/mysql/mysql.conf.d/mysqld.cnf
sudo sed -i -e '$innodb_file_per_table = 1' /etc/mysql/mysql.conf.d/mysqld.cnf
sudo sed -i -e '$innodb_file_format = Barracuda' /etc/mysql/mysql.conf.d/mysqld.cnf

sudo service mysql restart

sudo mysql < /vagrant/db_moodle.sql

#instalación
sudo chmod -R 0777 /var/www/html/moodle
install_vars=" --lang=es \
               --chmod=2770 \
               --wwwroot="http://localhost:8080/moodle" \
               --dataroot=/var/moodledata \
               --dbtype=mysqli \
               --dbhost=localhost \
               --dbname=moodle \
               --dbuser=moodledude \
               --dbpass=passwordformoodledude \
               --prefix="mdl_" \
               --fullname="Moodle" \
               --shortname="Moodle" \
               --adminuser=admin \
               --adminpass=This.is.Th3.Admin.Passw0rd \
               --non-interactive \
               --agree-license \
               --allow-unstable"

sudo su www-data -s /bin/bash -c \
    "/usr/bin/php /var/www/html/moodle/admin/cli/install.php $install_vars"

sudo chmod -R 0755 /var/www/html/moodle

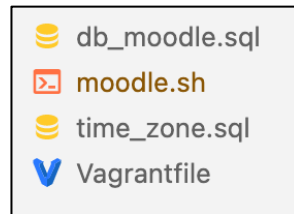
sudo service apache2 restart
```

Fuente: elaboración propia, empleando VS Code.

4.4. Unificación de las automatizaciones

Los *scripts* anteriores se unifican con el *script* de Vagrant y la estructura de los archivos queda de la siguiente manera, como se muestra en la figura 12.

Figura 12. **Estructura de archivos del proyecto Vagrant y *scripts* de instalación de Moodle**



Fuente: elaboración propia.

En el archivo Vagrantfile se define la automatización de la ejecución de los *scripts* que han sido definidos anteriormente, tal que, ejecutando el archivo moodle.sh como *script* con Vagrant se cree un flujo simple de configuración e instalación. Las propiedades del *script* del archivo Vagrantfile se muestran en la figura 13 y se explican a continuación:

- config.vm.box: define el sistema hashicorp/bionic64 que se utilizará para ejecutar la virtualización.
- config.vm.box_url: define el enlace de obtención del sistema de virtualización.
- config.vm.network: define la redirección de los puertos 80 de la máquina virtual con el puerto 8080 de la máquina física.
- config.vm.provision: define la ejecución del *script* de instalación principal de Moodle.

Figura 13. Configuración del proyecto Vagrant

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  config.vm.box = "hashicorp/bionic64"
  config.vm.box_url = "https://vagrantcloud.com/hashicorp/bionic64"
  config.vm.network "forwarded_port", guest: 80, host: 8080
  config.vm.provision :shell, path: "moodle.sh"
end
```

Fuente: elaboración propia, empleando VS Code.

Nuevamente, para ejecutar el proyecto se utiliza el comando de la figura 5, y con esto se concluye el proceso de automatización de un entorno replicable de la plataforma Moodle con el uso de Vagrant. En el capítulo siguiente se explica como esta herramienta facilitará la replicación de la plataforma base de *e-learning* Moodle de la RADD para ser utilizada por cualquier unidad académica de la Universidad.

5. CONFIGURACIÓN, INSTALACIÓN Y AUTOMATIZACIÓN VÍA VAGRANT DE LA PLATAFORMA *E-LEARNING* MOODLE

A continuación, se encontrarán los pasos por seguir para realizar la instalación y configuración de forma automatizada de la plataforma Moodle utilizando la herramienta de Vagrant. para que sirva como herramienta de apoyo en el desarrollo de futuras instalaciones y configuraciones requeridas.

Cabe mencionar, que las aulas virtuales de la RADD actualmente son instalaciones Moodle en su versión 3.8.2 (Build: 20200309) con modificaciones respecto de cualquier versión estándar. Las aulas virtuales RADD cuentan con cinco módulos, dos son de actividad (tarea y foro) y tres de recurso (carpeta, archivo y URL). Estos cambios se realizaron con el propósito de optimizar el recurso a nivel de servidor, y con esto obtener un mayor rendimiento en el servicio brindado a las unidades académicas alojadas en el sistema RADD.

5.1. Instalación de prerequisites para Moodle y configuraciones iniciales

Para la instalación de Moodle, se necesita tener los siguientes prerequisites instalados:

- PHP 7.2
- MySQL 8.0.20
- Apache 2.0

Además, se deben instalar las librerías necesarias para Moodle, las cuales son:

- php7.2-curl
- php7.2-zip
- php7.2-gd
- php7.2-intl
- php7.2-xmlrpc
- php7.2-soap
- *ghostscript*
- libgs-dev
- gs-esp

Por último, se debe configurar la zona horaria de la máquina utilizada, además de configurar ciertos parámetros en el `php.ini` que serán indicados más adelante.

Se debe correr el siguiente *script* que se muestra en la figura 14 para la instalación de los prerequisites, clonando el repositorio de Moodle con la versión 3.8 en caso se desee una versión limpia de Moodle o es posible utilizar una versión de esta, la cual ya haya sido personalizada.

Nota: algunas librerías pueden no ser necesarias para el servidor apache, sin embargo, sí serían útiles si se desea emplear otro servidor web.

Figura 14. **Script de instalación de prerequisites de Moodle**

```
#!/bin/bash
sudo apt-get update
#instalación de php 7.2
sudo apt-get install -y php7.2

##instalación de mysql
sudo apt-get install -y mysql-server mysql-client
#Definir hora local en el servidor
Date

#timedatectl list-timezones | grep -i america
sudo unlink /etc/localtime
sudo ln -s /usr/share/zoneinfo/America/Guatemala /etc/localtime
date

#Librerías necesarias de Moodle
sudo apt-get install -y php7.2-curl
sudo apt-get install -y php7.2-zip
sudo apt-get install -y php7.2-gd
sudo apt-get install -y php7.2-intl
sudo apt-get install -y php7.2-mysql
sudo apt-get install -y php7.2-ldap
sudo apt-get install -y php7.2-mbstring
sudo apt-get install -y php7.2-xmlrpc
sudo apt-get install -y php7.2-xml
sudo apt-get install -y php7.2-soap
sudo apt-get install -y ghostscript
sudo apt-get install -y libgs-dev
sudo apt-get install -y gs-es

#instalación de apache
sudo apt-get install -y apache2

#clonar repositorio de moodle
sudo git clone -
b MOODLE_38_STABLE git://git.moodle.org/moodle.git /var/www/html/moodle
```

Fuente: elaboración propia, empleando VS Code.

Posterior a la instalación de prerequisites se debe establecer la hora local en la base de datos.

Figura 15. **Definición de zona horaria de MYSQL**

```
#Definir hora local en la base de datos
mysql -u root -p
>SET time_zone = '-6:00';
>exit
service mysql restart
```

Fuente: elaboración propia, empleando VS Code.

Luego de establecer la hora local de la base de datos, se debe configurar los siguientes parámetros de php.ini

Figura 16. **Configuración de parámetros de php.ini**

```
#establecer limites
open_basedir = /home/{USERNAME}:/usr/bin/gs:/usr/bin/gs
upload_tmp_dir = /home/{USERNAME}/tmp
upload_max_filesize = 1M
post_max_size = 1M
```

Fuente: elaboración propia, empleando VS Code.

Se trabajará bajo la ruta /var/www/html, esta puede ser cambiada si así se desea, haciendo las modificaciones correspondientes en el servidor de apache o en el servidor que se esté empleando.

Figura 17. **Copia de repositorio a ruta deseada**

```
sudo mkdir /var/moodledata
sudo chown -R www-data /var/moodledata
sudo chmod -R 777 /var/moodledata
sudo chmod -R 0755 /var/www/html/Moodle
```

Fuente: elaboración propia, empleando VS Code.

5.2. Configuración de MYSQL y PHP

Como siguiente paso, se debe configurar MYSQL para lo cual se edita el archivo: /etc/mysql/mysql.conf.d/mysqld.cnf.

Figura 18. **Configuración de mysqld.conf**

```
#Estableceremos los siguientes parametros a la sección [mysqld]
default_storage_engine = innodb
innodb_file_per_table = 1
innodb_file_format = Barracuda

#Reiniciar el servicio mysql
sudo service mysql restart
```

Fuente: elaboración propia, empleando VS Code.

Como siguiente paso se debe ingresar a mysql, y ejecutar lo siguiente:

Figura 19. **Ejecución de sentencias en MYSQL**

```
CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;  
create user 'moodledude'@'localhost' IDENTIFIED BY 'passwordformoodledude';  
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,CREATE TEMPORARY  
TABLES,DROP,INDEX,ALTER ON moodle.* TO moodledude@localhost  
IDENTIFIED BY 'passwordformoodledude';
```

Fuente: elaboración propia, empleando VS Code.

Se debe eliminar el comentario o agregar la siguiente línea al archivo php.ini. en el caso del servidor apache la ruta sería la siguiente: /etc/php/7.2/apache2/php.ini.

Figura 20. **Configuración de php.ini**

```
extension = php_mysql
```

Fuente: elaboración propia, empleando VS Code.

5.3. **Culminación del proceso de instalación**

Para completar la instalación se deberá ejecutar el siguiente archivo.

Figura 21. **Ejecución de archivo de instalación install.php**

```
sudo -u www-data /usr/bin/php /var/www/html/moodle/admin/cli/install.php
```

Fuente: elaboración propia, empleando VS Code.

Por último, se debe asegurar que todos los archivos de Moodle tengan los permisos 755.

Figura 22. **Configuración de permisos**

```
sudo chmod -R 0755 /var/www/html/Moodle
```

Fuente: elaboración propia, empleando VS Code.

5.4. **Creación de entorno de Moodle replicable con Vagrant**

Debido a que el objetivo es automatizar la tarea de instalación de Moodle se utilizará Vagrant para crear un entorno reproducible, haciendo uso de archivos de configuración, por lo tanto, para continuar con este proceso previamente se debe haber realizado la instalación de Vagrant.

Las configuraciones que se realizaron anteriormente ahora estarán en un archivo llamado moodle.sh el cual podrá ser editado para su posterior uso, con los nombres de carpetas, base de datos, contraseña, entre otras configuraciones de Moodle. Además, este script hace uso de dos archivos más .sql, los cuales también podrán ser editados según se requiera.

A continuación, se explicará las diferentes secciones del *script*, entre las cuales se encontrarán:

- Comentarios que están marcados de color verde los cuales explican que realiza esa porción del *script*.
- Secciones del *script* marcadas de color naranja lo que significa que estos valores pueden ser modificables por los que se deseen.

- Secciones del *script* marcadas de color rojo lo que significa que estos valores serán modificados, según la ruta en donde se ubiquen los archivos para la automatización de la instalación.

Figura 23. **Script parametrizado de la instalación de Moodle y sus prerequisites**

```
#!/bin/bash
sudo apt-get update
#instalación de php 7.2
sudo apt-get install -y php7.2

##instalación de mysql
sudo apt-get install -y mysql-server mysql-client
#Definir hora local en el servidor
date

#timedatectl list-timezones | grep -i america
sudo unlink /etc/localtime
sudo ln -s /usr/share/zoneinfo/America/Guatemala /etc/localtime
date

#Librerías necesarias de Moodle
sudo apt-get install -y php7.2-curl
sudo apt-get install -y php7.2-zip
sudo apt-get install -y php7.2-gd
sudo apt-get install -y php7.2-intl
sudo apt-get install -y php7.2-mysql
sudo apt-get install -y php7.2-ldap
sudo apt-get install -y php7.2-mbstring
sudo apt-get install -y php7.2-xmlrpc
sudo apt-get install -y php7.2-xml
sudo apt-get install -y php7.2-soap
sudo apt-get install -y ghostscript
sudo apt-get install -y libgs-dev
sudo apt-get install -y gs-es
#instalación de apache
sudo apt-get install -y apache2
```


Continuación de la figura 23.

```
#clonar repositorio de moodle
# Instancia de Moodle limpia -> utilizar esta parte del script
si se desea realizar una nueva instalación desde el repositorio de
Moodle
sudo git clone -b MOODLE_38_STABLE git://git.moodle.org/moodle.git
/var/www/html/Moodle

sudo mkdir /var/moodledata
# Si se desea utilizar los archivos existentes de Moodle de una plataforma
que ya ha sido personalizada anteriormente, se deben descomentar la
siguientes líneas.
#sudo cp -r /vagrant/moodle /var/www/html/moodle
#sudo cp -r /vagrant/moodledata /var/moodledata

# Permisos de directorios
sudo chown -R www-data /var/moodledata
sudo chmod -R 777 /var/moodledata
sudo chmod -R 0755 /var/www/html/moodle

# Configurar mysql para UTC -6:00
sudo mysql < /vagrant/time_zone.sql

# Configurar servidor php -> estas configuraciones dependen de las rutas
que se desean en la maquina
#sudo sed -i -e '$aopen_basedir = /home/vagrant:/usr/bin/gs:/usr/bin/gs'
/etc/php/7.2/apache2/php.ini
#sudo sed -i -e '$aupload_tmp_dir = /home/vagrant/tmp'
/etc/php/7.2/apache2/php.ini
#sudo sed -i -e '$aupload_max_filesize = 1M' /etc/php/7.2/apache2/php.ini
#sudo sed -i -e '$apost_max_size = 1M' /etc/php/7.2/apache2/php.ini

sudo sed -i -e '$aextension = php_mysqli' /etc/php/7.2/apache2/php.ini

# Configurar mysql
sudo sed -i -e '$adefault_storage_engine = innodb'
/etc/mysql/mysql.conf.d/mysqld.cnf
sudo sed -i -e '$ainnodb_file_per_table = 1'
/etc/mysql/mysql.conf.d/mysqld.cnf
sudo sed -i -e '$ainnodb_file_format = Barracuda'
```

Continuación de la figura 23.

```
/etc/mysql/mysql.conf.d/mysqld.cnf

sudo service mysql restart

sudo mysql < /vagrant/db_moodle.sql

#instalación, en esta parte se configuran las variables de instalación, ta
les como la carpeta de MoodleData, el nombre de la base de datos, el usuar
io y contraseña de la base de datos, la base de datos a utilizar (mysql, p
ostgres, etc), entre otras configuraciones.
sudo chmod -R 0777 /var/www/html/moodle
install_vars="  --lang=es \
                --chmod=2770 \
                --wwwroot="http://localhost:8080/moodle" \
                --dataroot=/var/moodledata \
                --dbtype=mysqli \
                --dbhost=localhost \
                --dbname=moodle \
                --dbuser=moodledude \
                --dbpass=passwordformoodledude \
                --prefix="mdl_" \
                --fullname="Moodle" \
                --shortname="Moodle" \
                --adminuser=admin \
                --adminpass=Th1s.1s.Th3.Adm1n.Passw0rd \
                --non-interactive \
                --agree-license \
                --allow-unstable"

# Se ejecuta el script de instalación con las variables definidas
anteriormente
sudo su www-data -s /bin/bash -c \
    "/usr/bin/php /var/www/html/moodle/admin/cli/install.php $install_vars"

# Se cambian los permisos de los archivos
sudo chmod -R 0755 /var/www/html/moodle

# Se reiniciar el servidor de apache para poder visualizar los cambios
sudo service apache2 restart
```

Fuente: elaboración propia, empleando VS Code.

El archivo de creación de la base de datos de Moodle (db_moodle.sql), al igual que el archivo Moodle.sh, puede ser configurado según los valores que se requieran, este *script* es ejecutado desde Moodle.sh.

Figura 24. **Configuración de la base de datos**

```
CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci;

CREATE USER 'moodledude'@'localhost' IDENTIFIED BY 'passwordformoodledude';

GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,CREATE TEMPORARY
TABLES,DROP,INDEX,ALTER ON moodle.*
TO 'moodledude'@'localhost'
IDENTIFIED BY 'passwordformoodledude';
```

Fuente: elaboración propia, empleando VS Code.

Por último, el archivo de *time_zone.sql*, el cual establece la hora local en la base de datos:

Figura 25. **Definición de zona horario de la base de datos**

```
SET time_zone = '-6:00';
```

Fuente: elaboración propia, empleando VS Code.

Para ejecutar estos archivos es necesario tener Vagrant instalado, y en la carpeta en donde se ubiquen se encontrará un archivo llamado Vagrantfile, el cual cuenta con la configuración necesaria para ejecutar la instalación.

Figura 26. **Archivo de configuración de Vagrant**

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
#Configuración del Hipervisor, Seteado en 2 para VirtualBox
Vagrant.configure("2") do |config|
#Se define la imagen de la máquina virtual a utilizar (El valor aquí debe ser
  el nombre de un maquina instalada o un nombre abreviado de un maquina en Has
  hiCorp's Vagrant Cloud)-
  config.vm.box = "hashicorp/bionic64"
  #Se define la URL donde se puede encontrar la maquina configurada (Si config
  .vm.box es una forma abreviada de un cuadro en la nube de HashiCorp, entonces
  no es necesario especificar este valor. De lo contrario, debe apuntar al lug
  ar adecuado donde se puede encontrar la box si no está instalada. Esto tambié
  n puede ser una matriz de múltiples URL. Las URL se probarán en orden)
  config.vm.box_url = "https://vagrantcloud.com/hashicorp/bionic64"
  #Se define el nombre de la red virtual, el puerto a exponer desde la maquina
  virtual huesped, y el puerto mapeable desde el host
  config.vm.network "forwarded_port", guest: 80, host: 8080
  #Configura los aprovisionadores en la máquina, de modo que el software se pu
  ede instalar y configurar automáticamente cuando se crea la máquina.
  config.vm.provision :shell, path: "moodle.sh"
end
```

Fuente: elaboración propia, empleando VS Code.

Para iniciar la instalación se debe escribir en la terminal el siguiente comando:

Figura 27. **Comando para iniciar la instalación**

```
$ vagrant up
```

Fuente: Vagrant – Up. *Comendo*. www.vagrantup.com/docs/cli/up. Consulta: 15 de agosto de 2021.

Otra forma de realizar la instalación es ejecutando directamente el *script* Moodle.sh, pero cabe mencionar que, al realizar la instalación de esta forma, no se estará haciendo uso de la herramienta Vagrant y la instalación será realizada en la máquina *host* donde sea ejecutado este *script*, por lo tanto, se recomienda evaluar la forma en que se desea realizar la instalación.

CONCLUSIONES

1. Vagrant permite realizar el despliegue de entornos de desarrollo virtuales, y estos funcionan perfectamente para ambientes productivos que se desee sean reproducibles mediante archivos de configuración, permitiendo su replicación de forma automática.
2. La documentación del proceso de automatización realizada con la herramienta de Vagrant permitirá transmitir el conocimiento y facilitar la replicabilidad de los entornos virtuales de la plataforma base de la Red de Apoyo Digital Docente para ser utilizada por los centros universitarios, facultades y unidades académicas que así lo requieran.
3. Como parte de la escalabilidad y flexibilidad al cambio que toda plataforma, automatización o infraestructura en informática necesita, el uso de Vagrant permite migrar sin inconveniente el cambio de servicios en la nube, como por ejemplo a cualquiera de las marcas más representativas en la actualidad, Amazon Web Services, Google Cloud, Digital Ocean, y otras.

RECOMENDACIONES

1. Lograr que las casas de estudio de nivel superior puedan promover y favorecer de forma amplia y marcada el proceso de la cultura digital y educación virtual para el personal docente y estudiantes, más allá de las circunstancias provocadas por la pandemia.
2. Propiciar que los estudiantes consigan su transformación digital pero primero alcancen una transformación personal. En el tema educativo esta es fundamentalmente necesaria para facilitar el proceso de aprendizaje y así potencializar el uso de las herramientas que se vean involucradas, sin importar el factor distancia.
3. Concientizar a las unidades académicas para que motiven a sus docentes a fin de que tengan un acercamiento al uso de las herramientas que se utilicen, si bien su conocimiento puede ser no demasiado amplio, las nociones básicas son fundamentales para maximizar el aprendizaje en los alumnos.

BIBLIOGRAFÍA

1. Terry Slattery. *Ansible vs. Terraform vs. Vagrant. What's the difference?* [en línea]. <<https://www.techtarget.com/searchnetworking/tip/Ansible-vs-Terraform-vs-Vagrant-Whats-the-difference>>. [Consulta: 11 de septiembre de 2021].
2. DEDEV. *División de Educación a Distancia en Entornos Virtuales*. [en línea]. <<https://virtual.usac.edu.gt/>>. [Consulta: 14 de agosto de 2021].
3. Comunidad Moodle. *Guía rápida de Instalación*. [en línea]. <https://docs.moodle.org/all/es/Gu%C3%ADa_r%C3%A1pida_de_Instalaci%C3%B3n>. [Consulta: 15 de agosto de 2021].
4. IBM Cloud Education. *LAMP Stack*. [en línea]. <<https://www.ibm.com/cloud/learn/lamp-stack-explained>>. [Consulta: 14 de agosto de 2021].
5. Comunidad Moodle. *Notas de Moodle 3.8*. [en línea]. <https://docs.moodle.org/all/es/Notas_de_Moodle_3.8>. [Consulta: 14 de agosto de 2021].
6. RADD. *Red de Apoyo Digital Docente*. [en línea]. <<https://radd.virtual.usac.edu.gt/index.html>>. [Consulta: 14 de agosto de 2021].

7. HashiCorp. *Terraform Documentation*. [en línea].
<<https://www.terraform.io/docs/index.html>>. [Consulta: 11 de septiembre de 2021].
8. HashiCorp. *Vagrant Documentation*. [en línea].
<<https://www.vagrantup.com/docs/index>>. [Consulta: 15 de agosto de 2021].
9. VILLAGRA PACHECO, Joaquín Ignacio. *VAGRANT - Entornos de trabajo independientes, replicables y elegantes*. [en línea].
<<https://medium.com/@joaquin.villagra/vagrant-entornos-de-trabajo-independientes-replicables-y-elegantes-e49597eeeb65>>. [Consulta: 11 de septiembre de 2021].