**UNIVERSIDAD INTERNACIONAL DE LA RIOJA (UNIR)**  
*La Universidad en Internet*

**Herramientas de Automatización de Despliegues**

**Enero 2024**

Actividad: Empleo del Chef Workstation (grupal)

Equipo: 2\_H

Índice

[Introducción 3](#_Toc188109943)

[Cookbooks planteados 3](#_Toc188109944)

[Vagrantfile 4](#_Toc188109945)

[Configuración del Vagrantfile 4](#_Toc188109946)

[Estructura de los Cookbooks 5](#_Toc188109947)

[Cookbook de Apache y PHP 5](#_Toc188109948)

[Cookbook de MySQL 7](#_Toc188109949)

[Cookbook de Jenkins 8](#_Toc188109950)

[Creación de tests 9](#_Toc188109951)

[Unit test 9](#_Toc188109952)

[Integration Tests 11](#_Toc188109953)

[Correr test de integración 12](#_Toc188109954)

[Extra 16](#_Toc188109955)

[Resultados finales 17](#_Toc188109956)

[Comprobación Apache + PHP 26](#_Toc188109957)

[Comprobación MySQL 27](#_Toc188109958)

[Comprobación Jenkins 27](#_Toc188109959)

[Dificultades 29](#_Toc188109960)

[Conclusiones 30](#_Toc188109961)

[Referencias 31](#_Toc188109962)

# Introducción

El presente proyecto tiene como objetivo principal adquirir habilidades prácticas en el uso de Chef Workstation para la automatización de despliegues. A través de la instalación, configuración y uso de Chef Workstation, se busaca automatizar tareas de configuración y uso de Chef Workstation, se busca automatizar tareas de configuración mediante la creación de cookbooks y recetas. Para lograr este objetivo, se implementaron cookbooks que permiten la instalación y configuración de servicios esenciales como Apache con PHP, MySQL y Jenkins, utilizando Vagrant como herramienta de virtualización. Además, se incluyeron pruebas unitarias e integrales para validar el correcto funcionamiento de los despliegues, asegurando así la calidad y confiabilidad de las configuraciones automatizadas.

Dentro del desarrollo de esta actividad grupal, se enfrentaron diversos desafíos relacionados con la instalación de software de diferentes sistemas operativos, la configuración de Vagrantfile, la gestión de recursos limitados en las máquinas virtuales y la correcta configuración de puertos.

Repositorio con el código del proyecto:  
<https://github.com/luismendezc/unir/tree/main/1_semestre/Herramientas_de_Automatizacion_de_Despliegues>

# Cookbooks planteados

1. Apache y PHP: configuración de un servidor con Apache y PHP para mostrar una página web básica.

2. MySQL: configuración de una base de datos MySQL.

3. Jenkins: configuración de Jenkins para automatizar tareas de CI/CD

Como parte de los trabajos previos necesarios se instalará chef Workstation para Windows con la ayuda de Chocolatey y también instalaremos Vagrant y virtualbox.

* <https://chocolatey.org/install#individual>
* <https://docs.chef.io/workstation/install_workstation/>
* <https://www.vagrantup.com/>
* <https://www.virtualbox.org/>

# Vagrantfile

Con la ayuda de Vagrant crearemos un Vagrantfile con el siguiente comando:

vagrant init

## Configuración del Vagrantfile

Usaremos una máquina virtual que sea equivalente a ubuntu 20.04, en el caso del proveedor de máquinas virtuales usaremos virtual box previamente instaldo al igual que incrementaremos la memoria y número de cpu, esto debido a que la instalación de Jenkins demanda más requerimientos lo cual si no se agregar más capacidad puede que no funcioné, haremos una redirección de puertos y en el caso de MySQL si encuentra que el puerto ya esta siendo usado entonces puede auto corregirse y utilizar un puerto disponible, se sincronizarán la carpeta de cookbooks con la máquina que vagrant creará, y por último la configuración de chef, lo haremos de manera “solo” lo cual va a permitir a la máquina virtual instalar y aceptar la licencia de chef para poder ejecutar las recetas de nuestros cookbooks en un orden específico.

Vagrant.configure("2") do |config|

  # Seleccionamos la caja oficial de Ubuntu 20.04

  config.vm.box = "ubuntu/focal64"

  # Aumentar memoria y CPUs para que Jenkins funcione correctamente

  config.vm.provider "virtualbox" do |vb|

    vb.memory = 2048

    vb.cpus = 2

  end

  # Configuración de red: reenvío de puertos

  config.vm.network "forwarded\_port", guest: 80,   host: 8080   # Apache/PHP

  config.vm.network "forwarded\_port", guest: 3306, host: 3306, auto\_correct: true  # MySQL

  config.vm.network "forwarded\_port", guest: 8080, host: 8081   # Jenkins

  # Sincronización de carpetas: para que cookbooks estén dentro de la VM

  config.vm.synced\_folder "./cookbooks", "/home/vagrant/cookbooks"

  # Provisión con Chef Solo (incluimos las recetas en el run\_list)

  config.vm.provision "chef\_solo" do |chef|

    chef.cookbooks\_path = "./cookbooks"

    # Aquí decidimos en qué orden se aplicarán los cookbooks/recetas

    chef.add\_recipe "apache\_php::default"

    chef.add\_recipe "mysql\_server::default"

    chef.add\_recipe "jenkins::default"

    # Acepta la licencia Chef automáticamente

    chef.arguments = "--chef-license accept"

  end

end

# Estructura de los Cookbooks

Se creará un directorio denominado cookbooks el cual contendrá cada uno de nuestros cookbooks con sus respectivas configuraciones, para esto usaremos generate cookbook por parte de chef.

> mkdir cookbooks

cd cookbooks

chef generate cookbook apache\_php

chef generate cookbook mysql\_server

chef generate cookbook jenkins

Captura de pantalla de computadora

Descripción generada automáticamente

Figura 1. Estructura del proyecto en Visual Studio Code.

# Cookbook de Apache y PHP

Dentro del cookbook de apache\_php previamente generado crear recipes/apache.rb y agregar el siguiente contenido para poder tener el servidor de apache listo.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | *# Instala Apache*  package 'apache2' **do**  action :install  **end**  *# Habilita y levanta el servicio de Apache*  service 'apache2' **do**  action [:enable, :start]  **end**    *# Opcional: Elimina el index.html predeterminado*  file '/var/www/html/index.html' **do**  action :delete  **end** |

Del mismo modo crear la receta para php (recipes/php.rb):

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | *# Actualizar apt*  execute 'apt\_update' **do**  command 'apt-get update'  action :run  **end**  *# Instala PHP y el módulo Apache para interpretar PHP*  package 'php' **do**  action :install  **end**  package 'libapache2-mod-php' **do**  action :install  **end**  *# Reinicia Apache para reconocer PHP*  service 'apache2' **do**  action :restart  **end**  *# Crea un index.php con una página php sencilla*  file '/var/www/html/index.php' **do**  content '<?php phpinfo(); ?>'  action :create  **end** |

Actualizamos el archivo metadata.rb para poder especificar más información con respecto a este cookbook.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | name 'apache\_php'  maintainer 'The Authors'  maintainer\_email 'you@example.com'  license 'All Rights Reserved'  description 'Instala y configura Apache y PHP'  long\_description 'Instala/Configura apache\_php con apache y php'  version '0.1.0'  chef\_version '>= 14.0' |

Debido a que modularizamos nuestro cookbook entonces por eso tendremos que incluir nuestras recetas dentro de la receta default.rb:

|  |  |
| --- | --- |
| 1  2 | include\_recipe 'apache\_php::apache'  include\_recipe 'apache\_php::php' |

# Cookbook de MySQL

Receta de MySQL:

Dentro del cookbook de mysql\_server previamente generado crear recipes/mysql.rb para poder instalar e iniciar mysql, utilizaremos un archivo de configuración files/default/mysqld.cnf

[mysqld]

bind-address = 0.0.0.0

Para poder hacer uso de MySQL desde la máquina Windows creamos un usuario denominado myuser con contraseña mypassword y por último crearemos una base de datos con el nombre mydatabase:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | *# Instala MySQL Server*  package 'mysql-server' **do**  action :install  **end**  *# Activa el servicio*  service 'mysql' **do**  action [:enable, :start]  **end**  *# Ajustamos la configuración para escuchar en 0.0.0.0*  cookbook\_file '/etc/mysql/mysql.conf.d/mysqld.cnf' **do**  source 'mysqld.cnf'  owner 'root'  group 'root'  mode '0644'  notifies :restart, 'service[mysql]', :immediately  **end**  *# Crear el usuario remoto*  execute 'create\_remote\_user' **do**  command <<-EOF  mysql -u root -e "CREATE USER IF NOT EXISTS 'myuser'@'%' IDENTIFIED BY 'mypassword';  GRANT ALL PRIVILEGES ON \*.\* TO 'myuser'@'%' WITH GRANT OPTION;  FLUSH PRIVILEGES;"  EOF  action :run  **end**  *# Crear la base de datos*  execute 'create\_db' **do**  command 'mysql -u root -e "CREATE DATABASE IF NOT EXISTS mydatabase"'  action :run  **end** |

Modificamos nuestro archivo de metadatos.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | name 'mysql\_server'  maintainer 'The Authors'  maintainer\_email 'you@example.com'  license 'All Rights Reserved'  description 'Instala y configura MySQL'  long\_description 'Instala/Configura mysql\_server con usuario y base de datos'  version '0.1.0'  chef\_version '>= 14.0' |

Debido a la modularización de nuestro cookbook tendremos que incluir nuestra receta dentro de la receta default.rb:

|  |  |
| --- | --- |
| 1 | include\_recipe 'mysql\_server::mysql' |

# Cookbook de Jenkins

En el caso de Jenkins debido a que solo necesitaremos una receta lo mantendremos todo dentro de default.rb dentro del cookbook jenkins previamente generado.

Instalaremos los paquetes necesarios para que Jenkins pueda funcionar, java 11 y net-tools, al igual que el paquete .deb más estable de Jenkins para poder así instalarlo:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | *# Instalar Java (requerido por Jenkins)*  package 'openjdk-11-jdk' **do**  action :install  **end**  *# Instalar dependencias básicas que Jenkins necesita*  package 'net-tools' **do**  action :install  **end**  *# Descargar el paquete .deb de Jenkins (versión estable elegida 2.387.2)*  remote\_file '/tmp/jenkins.deb' **do**  source 'https://pkg.jenkins.io/debian-stable/binary/jenkins\_2.387.2\_all.deb'  action :create  **end**  *# Instalar Jenkins usando el archivo .deb*  dpkg\_package 'jenkins' **do**  source '/tmp/jenkins.deb'  action :install  **end**  *# Iniciar y habilitar Jenkins*  service 'jenkins' **do**  action [:enable, :start]  **end** |

Actualizar nuestro archivo de metadatos.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | name 'jenkins'  maintainer 'The Authors'  maintainer\_email 'you@example.com'  license 'All Rights Reserved'  description 'Instala y configura Jenkins'  long\_description 'Installa/Configurea jenkins a partir de .deb 2.387.2'  version '0.1.0'  chef\_version '>= 14.0' |

# Creación de tests

La creación de test será de gran ayuda ya que nos va a permitir estar seguros de que la implementación es correcta, para ello haremos test unit y de integración.

## Unit test

Estos test probaran funcionalidades específicas dentro de nuestro cookbook para comprobar si una funcionalidad está implementada de manera correcta por sí misma.

Dentro de la ruta spec/unit/recipes de cada cookbook podemos modificar default\_spec.rb, para ello configuramos el contexto en este caso Ubuntu 20.04 que tendrá SoloRunner para hacer uso de chef, se hace la validación de diferentes aspectos como instalación, estado de los servicios, creación y eliminación de archivos.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53 | *#*  *# Cookbook:: apache\_php*  *# Spec:: default*  *#*  *# Copyright:: 2025, The Authors, All Rights Reserved.*  require 'spec\_helper'  describe 'apache\_php::default' **do**  context 'En Ubuntu 20.04' **do**  let(:chef\_run) **do**  *# Simula un run de Chef en Ubuntu 20.04*  ChefSpec::SoloRunner.new(platform: 'ubuntu', version: '20.04')  .converge(described\_recipe)  **end**  *# Verifica la instalación de Apache*  it 'instala el paquete apache2' **do**  expect(chef\_run).to install\_package('apache2')  **end**  *# Verifica que habilita y arranca el servicio de Apache*  it 'habilita y arranca el servicio apache2' **do**  expect(chef\_run).to enable\_service('apache2')  expect(chef\_run).to start\_service('apache2')  **end**  *# Verifica que elimina el index.html si existe*  it 'elimina el index.html por defecto' **do**  expect(chef\_run).to delete\_file('/var/www/html/index.html')  **end**  *# Verifica la instalación de PHP*  it 'instala el paquete php' **do**  expect(chef\_run).to install\_package('php')  **end**  it 'instala el paquete libapache2-mod-php' **do**  expect(chef\_run).to install\_package('libapache2-mod-php')  **end**  *# Verifica que reinicia apache2 tras instalar mod-php*  it 'reinicia el servicio apache2' **do**  expect(chef\_run).to restart\_service('apache2')  **end**  *# Verifica que crea el archivo index.php con phpinfo()*  it 'crea el archivo /var/www/html/index.php con contenido phpinfo()' **do**  expect(chef\_run).to create\_file('/var/www/html/index.php')  .with\_content('<?php phpinfo(); ?>')  **end**  **end**  **end** |

Para comprobar el resultado necesitamos correr el siguiente comando dentro del cookbook con los unit test:

PS D:\UNIR\Maestria\unir\1\_semestre\Herramientas\_de\_Automatizacion\_de\_Despliegues\cookbooks\apache\_php> chef exec rspec spec

.......

Finished **in** 3.33 seconds (files took 31.01 seconds to load)

7 examples, 0 failures

Integration Tests

Antes de poder desplegar nuestros resultados el uso de tests de integración nos ayuda a comprobar si los elementos definidos en conjunto dentro del cookbook funcionan de manera correcta, esto hará uso de Test Kitchen una herramienta que nos ayuda a crear una máquina virtual, implementar nuestro cookbook y hacer las verificaciones que nosotros especifiquemos.

Necesitamos crear un archivo denominado kitchen.yml con el driver (Vagrant), el aprovisionador (chef), verificador, plataforma, etc.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | *# .kitchen.yml*  **---**  driver:  name: vagrant  provisioner:  name: chef\_zero  product\_name: chef  product\_version: latest  verifier:  name: inspec  platforms:  - name: ubuntu-20.04  suites:  - name: default  run\_list:  - recipe[apache\_php::default]  verifier:  inspec\_tests:  - test/integration/default  attributes: |

Antes de hacer las pruebas es necesario escribir las validaciones que queremos hacer, para eso dentro de test/integration/default/default\_test.rb pondremos nuestras verificaciones:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | describe package('apache2') **do**  it { should be\_installed }  **end**  describe service('apache2') **do**  it { should be\_enabled }  it { should be\_running }  **end**  describe package('php') **do**  it { should be\_installed }  **end**  describe package('libapache2-mod-php') **do**  it { should be\_installed }  **end**  describe file('/var/www/html/index.html') **do**  it { should\_not exist }  **end**  describe file('/var/www/html/index.php') **do**  it { should exist }  its('content') { should match /phpinfo/ }  **end** |

### Correr test de integración

Creamos la máquina donde los tests van a correr, esto iniciará una máquina virtual Vagrant con Ubuntu 20.04 y preparará el entorno, pero aún no aplicará el libro de recetas de Chef:

PS D:**\U**NIR**\M**aestria**\u**nir**\1**\_semestre**\H**erramientas\_de\_Automatizacion\_de\_Despliegues**\c**ookbooks**\a**pache\_php> kitchen create

-----> Starting Test Kitchen (v3.6.0)

-----> Creating <default-ubuntu-2004>...

Bringing machine 'default' up with 'virtualbox' provider...

==> default: Importing base box 'bento/ubuntu-20.04'...

==> default: Matching MAC address **for** NAT networking...

==> default: Checking **if** box 'bento/ubuntu-20.04' version '202407.23.0' is up to date...

==> default: Setting the name of the VM: kitchen-apache\_php-default-ubuntu-2004-701cce5e-3bcf-4980-a61b-421d8a43074a

==> default: Clearing any previously set network interfaces...

==> default: Preparing network interfaces based on configuration...

default: Adapter 1: nat

==> default: Forwarding ports...

default: 22 (guest) => 2222 (host) (adapter 1)

==> default: Running 'pre-boot' VM customizations...

==> default: Booting VM...

==> default: Waiting **for** machine to boot. This may take a few minutes...

default: SSH address: 127.0.0.1:2222

default: SSH username: vagrant

default: SSH auth method: private key

default:

default: Vagrant insecure key detected. Vagrant will automatically replace

default: this with a newly generated keypair **for** better security.

default:

default: Inserting generated public key within guest...

default: Removing insecure key from the guest **if** it's present...

default: Key inserted! Disconnecting and reconnecting using new SSH key...

==> default: Machine booted and ready!

==> default: Checking **for** guest additions in VM...

default: The guest additions on this VM **do** not match the installed version of

default: VirtualBox! In most cases this is fine, but in rare cases it can

default: prevent things such as shared folders from working properly. If you see

default: shared folder errors, please make sure the guest additions within the

default: virtual machine match the version of VirtualBox you have installed on

default: your host and reload your VM.

default:

default: Guest Additions Version: 7.0.18

default: VirtualBox Version: 7.1

==> default: Setting hostname...

==> default: Mounting shared folders...

default: C:/Users/lemen/.kitchen/cache => /tmp/omnibus/cache

==> default: Machine not provisioned because `--no-provision` is specified.

[SSH] Established

Vagrant instance <default-ubuntu-2004> created.

Finished creating <default-ubuntu-2004> (1m39.26s).

-----> Test Kitchen is finished. (1m49.73s)

Converge el libro de recetas, esto aplicará la receta apache\_php::default e instalará Apache y PHP, configurará los servicios y configurará los archivos según las recetas.

PS D:**\U**NIR**\M**aestria**\u**nir**\1**\_semestre**\H**erramientas\_de\_Automatizacion\_de\_Despliegues**\c**ookbooks**\a**pache\_php> kitchen converge

-----> Starting Test Kitchen (v3.6.0)

-----> Converging <default-ubuntu-2004>...

Preparing files **for** transfer

Installing cookbooks **for** Policyfile D:/UNIR/Maestria/unir/1\_semestre/Herramientas\_de\_Automatizacion\_de\_Despliegues/cookbooks/apache\_php/Policyfile.rb using `C:**\o**pscode**\c**hef-workstation**\b**in**\c**hef-cli.BAT install`

Installing cookbooks from lock

Installing apache\_php 0.1.0

Updating policy lock using `C:**\o**pscode**\c**hef-workstation**\b**in**\c**hef-cli.BAT update`

Building policy apache\_php

Expanded run list: recipe[apache\_php::default]

Caching Cookbooks...

Installing apache\_php >= 0.0.0 from path

Lockfile written to D:/UNIR/Maestria/unir/1\_semestre/Herramientas\_de\_Automatizacion\_de\_Despliegues/cookbooks/apache\_php/Policyfile.lock.json

Policy revision id: 1ee5b3dfeb3313078e0fd724a9319df4825be0e59c9c3de6fadc1bbaf292502a

Preparing dna.json

Exporting cookbook dependencies from Policyfile C:/Users/lemen/AppData/Local/Temp/default-ubuntu-2004-sandbox-20250118-26244-1mhcbi using `C:**\o**pscode**\c**hef-workstation**\b**in**\c**hef-cli.BAT export`...

Exported policy 'apache\_php' to C:/Users/lemen/AppData/Local/Temp/default-ubuntu-2004-sandbox-20250118-26244-1mhcbi

To converge this system with the exported policy, run:

cd C:/Users/lemen/AppData/Local/Temp/default-ubuntu-2004-sandbox-20250118-26244-1mhcbi

chef-client -z

Removing non-cookbook files before transfer

Preparing validation.pem

Preparing client.rb

ubuntu 20.04 x86\_64

Getting information **for** chef stable latest **for** ubuntu...

downloading https://omnitruck.chef.io/stable/chef/metadata?v=latest&p=ubuntu&pv=20.04&m=x86\_64

to file /tmp/kitchen/metadata.txt

trying wget...

sha1 d340c180bdb82c3b67145fdb0cd3ea38b1cd8308

sha256 8c0b4ede4d6996355297ae0c8423f34a7e5387e7bbf477aaca7a00a87effff6b

url https://packages.chef.io/files/stable/chef/18.6.2/ubuntu/20.04/chef\_18.6.2-1\_amd64.deb

version 18.6.2

downloaded metadata file looks valid...

/tmp/omnibus/cache/chef\_18.6.2-1\_amd64.deb exists

Comparing checksum with sha256sum...

Installing chef latest

installing with dpkg...

Selecting previously unselected package chef.

(Reading database ... 41439 files and directories currently installed.)

Preparing to unpack .../cache/chef\_18.6.2-1\_amd64.deb ...

Unpacking chef (18.6.2-1) ...

Setting up chef (18.6.2-1) ...

Thank you **for** installing Chef Infra Client! For help getting started visit https://learn.chef.io

Transferring files to <default-ubuntu-2004>

+---------------------------------------------+

✔ 2 product licenses accepted.

+---------------------------------------------+

Chef Infra Client, version 18.6.2

Patents: https://www.chef.io/patents

Infra Phase starting

Creating a new client identity **for** default-ubuntu-2004 using the validator key.

Using Policyfile 'apache\_php' at revision '1ee5b3dfeb3313078e0fd724a9319df4825be0e59c9c3de6fadc1bbaf292502a'

Resolving cookbooks **for** run list: ["apache\_php::default@0.1.0 (cdc6654)"]

Synchronizing cookbooks:

- apache\_php (0.1.0)

Installing cookbook gem dependencies:

Compiling cookbooks...

Loading Chef InSpec profile files:

Loading Chef InSpec input files:

Loading Chef InSpec waiver files:

Converging 8 resources

Recipe: apache\_php::apache

\* apt\_package[apache2] action install

- install version 2.4.41-4ubuntu3.21 of package apache2

\* service[apache2] action enable (up to date)

\* service[apache2] action start (up to date)

\* file[/var/www/html/index.html] action delete

- delete file /var/www/html/index.html

Recipe: apache\_php::php

\* execute[apt\_update] action run

- execute apt-get update

\* apt\_package[php] action install

- install version 2:7.4+75 of package php

\* apt\_package[libapache2-mod-php] action install

- install version 2:7.4+75 of package libapache2-mod-php

\* service[apache2] action restart

- restart service service[apache2]

\* file[/var/www/html/index.php] action create

- create new file /var/www/html/index.php

- update content in file /var/www/html/index.php from none to 075368

--- /var/www/html/index.php 2025-01-18 13:33:55.499614823 +0000

+++ /var/www/html/.chef-index20250118-13173-uifhef.php 2025-01-18 13:33:55.499614823 +0000

@@ -1 +1,2 @@

+<?php phpinfo(); ?>

Running handlers:

Running handlers complete

Infra Phase complete, 7/9 resources updated in 54 seconds

Downloading files from <default-ubuntu-2004>

Finished converging <default-ubuntu-2004> (2m0.26s).

-----> Test Kitchen is finished. (2m11.69s)

Ejecutar pruebas de integración, las pruebas InSpec que se encuentran en test/integration/default/default\_test.rb.

Verificará que Apache y PHP estén instalados, que los servicios se estén ejecutando y que los archivos estén configurados correctamente.

PS D:**\U**NIR**\M**aestria**\u**nir**\1**\_semestre**\H**erramientas\_de\_Automatizacion\_de\_Despliegues**\c**ookbooks**\a**pache\_php> kitchen verify

-----> Starting Test Kitchen (v3.6.0)

-----> Setting up <default-ubuntu-2004>...

Finished setting up <default-ubuntu-2004> (0m0.00s).

-----> Verifying <default-ubuntu-2004>...

Loaded tests from {:path=>"D:.UNIR.Maestria.unir.1\_semestre.Herramientas\_de\_Automatizacion\_de\_Despliegues.cookbooks.apache\_php.test.integration.default"}

Profile: tests from {:path=>"D:/UNIR/Maestria/unir/1\_semestre/Herramientas\_de\_Automatizacion\_de\_Despliegues/cookbooks/apache\_php/test/integration/default"} (tests from {:path=>"D:.UNIR.Maestria.unir.1\_semestre.Herramientas\_de\_Automatizacion\_de\_Despliegues.cookbooks.apache\_php.test.integration.default"})

Version: (not specified)

Target: ssh://vagrant@127.0.0.1:2222

Target ID: 506cc0f3-7c96-50cc-99d4-6eb14526f966

System Package apache2

[PASS] is expected to be installed

Service apache2

[PASS] is expected to be enabled

[PASS] is expected to be running

System Package php

[PASS] is expected to be installed

System Package libapache2-mod-php

[PASS] is expected to be installed

File /var/www/html/index.html

[PASS] is expected not to exist

File /var/www/html/index.php

[PASS] is expected to exist

[PASS] content is expected to match /phpinfo/

Test Summary: 8 successful, 0 failures, 0 skipped

Finished verifying <default-ubuntu-2004> (0m15.73s).

-----> Test Kitchen is finished. (0m27.30s)

### Extra

Para eliminar la máquina creada por los test de integración podemos usar el siguiente comando:

kitchen destroy

En el caso de no querer hacer la ejecución de los test de integración paso por paso Test Kitchen nos ofrece una manera más conveniente para poder ejecutar los tests (create → converge → verify → destroy):

kitchen test

# Resultados finales

Para desplegar nuestro proyecto nos iremos a la raíz del proyecto donde nuestro Vagrantfile se encuentra y ejecutaremos el siguiente comando:  
vagrant up

Alternativamente podemos agregar el proveedor, aunque ya está especificado dentro de nuestro Vagrantfile.

PS D:**\U**NIR**\M**aestria**\u**nir**\1**\_semestre**\H**erramientas\_de\_Automatizacion\_de\_Despliegues> vagrant up --provider=virtualbox

Bringing machine 'default' up with 'virtualbox' provider...

==> default: Importing base box 'ubuntu/focal64'...

==> default: Matching MAC address **for** NAT networking...

==> default: Checking **if** box 'ubuntu/focal64' version '20240821.0.1' is up to date...

==> default: Setting the name of the VM: Herramientas\_de\_Automatizacion\_de\_Despliegues\_default\_1737211046786\_46107

==> default: Auto-generating node name **for** Chef...

==> default: Fixed port collision **for** 3306 => 3306. Now on port 2200.

==> default: Clearing any previously set network interfaces...

==> default: Preparing network interfaces based on configuration...

default: Adapter 1: nat

==> default: Forwarding ports...

default: 80 (guest) => 8080 (host) (adapter 1)

default: 3306 (guest) => 2200 (host) (adapter 1)

default: 8080 (guest) => 8081 (host) (adapter 1)

default: 22 (guest) => 2222 (host) (adapter 1)

==> default: Running 'pre-boot' VM customizations...

==> default: Booting VM...

==> default: Waiting **for** machine to boot. This may take a few minutes...

default: SSH address: 127.0.0.1:2222

default: SSH username: vagrant

default: SSH auth method: private key

default:

default: Vagrant insecure key detected. Vagrant will automatically replace

default: this with a newly generated keypair **for** better security.

default:

default: Inserting generated public key within guest...

default: Removing insecure key from the guest **if** it's present...

default: Key inserted! Disconnecting and reconnecting using new SSH key...

==> default: Machine booted and ready!

==> default: Checking for guest additions in VM...

default: The guest additions on this VM do not match the installed version of

default: VirtualBox! In most cases this is fine, but in rare cases it can

default: prevent things such as shared folders from working properly. If you see

default: shared folder errors, please make sure the guest additions within the

default: virtual machine match the version of VirtualBox you have installed on

default: your host and reload your VM.

default:

default: Guest Additions Version: 6.1.50

default: VirtualBox Version: 7.1

==> default: Mounting shared folders...

default: D:/UNIR/Maestria/unir/1\_semestre/Herramientas\_de\_Automatizacion\_de\_Despliegues => /vagrant

default: D:/UNIR/Maestria/unir/1\_semestre/Herramientas\_de\_Automatizacion\_de\_Despliegues/cookbooks => /home/vagrant/cookbooks

default: D:/UNIR/Maestria/unir/1\_semestre/Herramientas\_de\_Automatizacion\_de\_Despliegues/cookbooks => /tmp/vagrant-chef/5de5ccee13edcb9211cfa1dbe1309d85/cookbooks

==> default: Running provisioner: chef\_solo...

default: Installing Chef (latest)...

==> default: Generating chef JSON and uploading...

==> default: Running chef-solo...

==> default: [2025-01-18T14:39:05+00:00] INFO: Persisting a license for Chef Infra Client at path /etc/chef/accepted\_licenses/chef\_infra\_client

==> default: [2025-01-18T14:39:05+00:00] INFO: Persisting a license for Chef InSpec at path /etc/chef/accepted\_licenses/inspec

==> default: +---------------------------------------------+

==> default: ✔ 2 product licenses accepted.

==> default: +---------------------------------------------+

==> default: [2025-01-18T14:39:05+00:00] INFO: Started Chef Infra Zero at chefzero://localhost:1 with repository at /tmp/vagrant-chef/5de5ccee13edcb9211cfa1dbe1309d85 (One version per cookbook)

==> default: Chef Infra Client, version 18.6.2

==> default: Patents: https://www.chef.io/patents

==> default: Infra Phase starting

==> default: [2025-01-18T14:39:05+00:00] INFO: \*\*\* Chef Infra Client 18.6.2 \*\*\*

==> default: [2025-01-18T14:39:05+00:00] INFO: Platform: x86\_64-linux

==> default: [2025-01-18T14:39:05+00:00] INFO: Chef-client pid: 2865

==> default: [2025-01-18T14:39:06+00:00] INFO: Setting the run\_list to ["recipe[apache\_php::default]", "recipe[mysql\_server::default]", "recipe[jenkins::default]"] from CLI options

==> default: [2025-01-18T14:39:06+00:00] INFO: Run List is [recipe[apache\_php::default], recipe[mysql\_server::default], recipe[jenkins::default]]

==> default: [2025-01-18T14:39:06+00:00] INFO: Run List expands to [apache\_php::default, mysql\_server::default, jenkins::default]

==> default: [2025-01-18T14:39:06+00:00] INFO: Starting Chef Infra Client Run for vagrant-ebedbd28

==> default: [2025-01-18T14:39:06+00:00] INFO: Running start handlers

==> default: [2025-01-18T14:39:06+00:00] INFO: Start handlers complete.

==> default: Resolving cookbooks for run list: ["apache\_php::default", "mysql\_server::default", "jenkins::default"]

==> default: [2025-01-18T14:39:07+00:00] INFO: Loading cookbooks [apache\_php@0.1.0, mysql\_server@0.1.0, jenkins@0.1.0]

==> default: Synchronizing cookbooks:

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/apache\_php/Vagrantfile.custom in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/apache\_php/recipes/php.rb in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/apache\_php/recipes/default.rb in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/apache\_php/LICENSE in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/mysql\_server/files/default/mysqld.cnf in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/apache\_php/README.md in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/mysql\_server/chefignore in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/apache\_php/metadata.rb in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/mysql\_server/LICENSE in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/apache\_php/recipes/apache.rb in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/mysql\_server/metadata.rb in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/apache\_php/chefignore in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/mysql\_server/recipes/default.rb in the cache.

==> default: - apache\_php (0.1.0)

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/mysql\_server/recipes/mysql.rb in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/jenkins/README.md in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/mysql\_server/README.md in the cache.

==> default: - mysql\_server (0.1.0)

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/jenkins/chefignore in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/jenkins/recipes/default.rb in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/jenkins/LICENSE in the cache.

==> default: [2025-01-18T14:39:07+00:00] INFO: Storing updated cookbooks/jenkins/metadata.rb in the cache.

==> default: - jenkins (0.1.0)

==> default: Installing cookbook gem dependencies:

==> default: Compiling cookbooks...

==> default: Loading Chef InSpec profile files:

==> default: Loading Chef InSpec input files:

==> default: Loading Chef InSpec waiver files:

==> default: Converging 18 resources

==> default: Recipe: apache\_php::apache

==> default: \* apt\_package[apache2] action install

==> default: [2025-01-18T14:39:24+00:00] INFO: apt\_package[apache2] installed apache2 at 2.4.41-4ubuntu3.21

==> default:

==> default: - install version 2.4.41-4ubuntu3.21 of package apache2

==> default: \* service[apache2] action enable

==> default: (up to date)

==> default: \* service[apache2] action start

==> default: (up to date)

==> default: \* file[/var/www/html/index.html] action delete

==> default: [2025-01-18T14:39:24+00:00] INFO: file[/var/www/html/index.html] backed up to /var/chef/backup/var/www/html/index.html.chef-20250118143924.651322

==> default: [2025-01-18T14:39:24+00:00] INFO: file[/var/www/html/index.html] deleted file at /var/www/html/index.html

==> default:

==> default: - delete file /var/www/html/index.html

==> default: Recipe: apache\_php::php

==> default:

==> default: \* execute[apt\_update] action run

==> default:

==> default: [execute] Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease

==> default: Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease

==> default: Hit:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease

==> default: Hit:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease

==> default: Reading package lists...

==> default: [2025-01-18T14:39:27+00:00] INFO: execute[apt\_update] ran successfully

==> default: - execute apt-get update

==> default: \* apt\_package[php] action install

==> default: [2025-01-18T14:39:45+00:00] INFO: apt\_package[php] installed php at 2:7.4+75

==> default:

==> default: - install version 2:7.4+75 of package php

==> default: \* apt\_package[libapache2-mod-php] action install

==> default: [2025-01-18T14:39:47+00:00] INFO: apt\_package[libapache2-mod-php] installed libapache2-mod-php at 2:7.4+75

==> default:

==> default: - install version 2:7.4+75 of package libapache2-mod-php

==> default: \* service[apache2] action restart

==> default: [2025-01-18T14:39:47+00:00] INFO: service[apache2] restarted

==> default:

==> default: - restart service service[apache2]

==> default: \* file[/var/www/html/index.php] action create

==> default: [2025-01-18T14:39:47+00:00] INFO: file[/var/www/html/index.php] created file /var/www/html/index.php

==> default:

==> default: - create new file /var/www/html/index.php

==> default: [2025-01-18T14:39:47+00:00] INFO: file[/var/www/html/index.php] updated file contents /var/www/html/index.php

==> default:

==> default: - update content in file /var/www/html/index.php from none to 075368

==> default:

==> default: --- /var/www/html/index.php 2025-01-18 14:39:47.778363466 +0000

==> default:

==> default: +++ /var/www/html/.chef-index20250118-2865-5q4w0i.php 2025-01-18 14:39:47.778363466 +0000

==> default:

==> default: @@ -1 +1,2 @@

==> default:

==> default: +<?php phpinfo(); ?>

==> default: Recipe: mysql\_server::mysql

==> default:

==> default: \* apt\_package[mysql-server] action install

==> default: [2025-01-18T14:40:43+00:00] INFO: apt\_package[mysql-server] installed mysql-server at 8.0.40-0ubuntu0.20.04.1

==> default:

==> default:

==> default: - install version 8.0.40-0ubuntu0.20.04.1 of package mysql-server

==> default:

==> default: \* service[mysql] action enable

==> default: (up to date)

==> default:

==> default: \* service[mysql] action start

==> default: (up to date)

==> default: \* cookbook\_file[/etc/mysql/mysql.conf.d/mysqld.cnf] action create

==> default: [2025-01-18T14:40:43+00:00] INFO: cookbook\_file[/etc/mysql/mysql.conf.d/mysqld.cnf] backed up to /var/chef/backup/etc/mysql/mysql.conf.d/mysqld.cnf.chef-20250118144043.541232

==> default: [2025-01-18T14:40:43+00:00] INFO: cookbook\_file[/etc/mysql/mysql.conf.d/mysqld.cnf] updated file contents /etc/mysql/mysql.conf.d/mysqld.cnf

==> default:

==> default: - update content in file /etc/mysql/mysql.conf.d/mysqld.cnf from 156886 to 79853e

==> default: --- /etc/mysql/mysql.conf.d/mysqld.cnf 2023-06-14 19:23:16.000000000 +0000

==> default: +++ /etc/mysql/mysql.conf.d/.chef-mysqld20250118-2865-i8wj1n.cnf 2025-01-18 14:40:43.536700948 +0000

==> default: @@ -1,79 +1,3 @@

==> default:

==> default: -#

==> default: -# The MySQL database server configuration file.

==> default:

==> default: -#

==> default: -# One can use all long options that the program supports.

==> default: -# Run program with --help to get a list of available options and with

==> default: -# --print-defaults to see which it would actually understand and use.

==> default: -#

==> default: -# For explanations see

==> default: -# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

==> default: -

==> default: -# Here is entries for some specific programs

==> default: -# The following values assume you have at least 32M ram

==> default: -

==> default: [mysqld]

==> default: -#

==> default: -# \* Basic Settings

==> default:

==> default: -#

==> default: -user = mysql

==> default: -# pid-file = /var/run/mysqld/mysqld.pid

==> default: -# socket = /var/run/mysqld/mysqld.sock

==> default: -# port = 3306

==> default: -# datadir = /var/lib/mysql

==> default:

==> default: -

==> default: -

==> default: -# If MySQL is running as a replication slave, this should be

==> default: -# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar\_tmpdir

==> default: -# tmpdir = /tmp

==> default: -#

==> default: -# Instead of skip-networking the default is now to listen only on

==> default: -# localhost which is more compatible and is not less secure.

==> default:

==> default: -bind-address = 127.0.0.1

==> default: -mysqlx-bind-address = 127.0.0.1

==> default:

==> default: -#

==> default: -# \* Fine Tuning

==> default:

==> default: -#

==> default: -key\_buffer\_size = 16M

==> default: -# max\_allowed\_packet = 64M

==> default: -# thread\_stack = 256K

==> default: -

==> default:

==> default: -# thread\_cache\_size = -1

==> default: -

==> default: -# This replaces the startup script and checks MyISAM tables if needed

==> default: -# the first time they are touched

==> default:

==> default: -myisam-recover-options = BACKUP

==> default: -

==> default: -# max\_connections = 151

==> default: -

==> default: -# table\_open\_cache = 4000

==> default: -

==> default: -#

==> default: -# \* Logging and Replication

==> default: -#

==> default: -# Both location gets rotated by the cronjob.

==> default: -#

==> default: -# Log all queries

==> default: -# Be aware that this log type is a performance killer.

==> default:

==> default: -# general\_log\_file = /var/log/mysql/query.log

==> default: -# general\_log = 1

==> default: -#

==> default:

==> default: -# Error log - should be very few entries.

==> default: -#

==> default: -log\_error = /var/log/mysql/error.log

==> default: -#

==> default: -# Here you can see queries with especially long duration

==> default:

==> default: -# slow\_query\_log = 1

==> default: -# slow\_query\_log\_file = /var/log/mysql/mysql-slow.log

==> default: -# long\_query\_time = 2

==> default:

==> default: -# log-queries-not-using-indexes

==> default: -#

==> default: -# The following can be used as easy to replay backup logs or for replication.

==> default: -# note: if you are setting up a replication slave, see README.Debian about

==> default: -# other settings you may need to change.

==> default: -# server-id = 1

==> default: -# log\_bin = /var/log/mysql/mysql-bin.log

==> default:

==> default: -# binlog\_expire\_logs\_seconds = 2592000

==> default: -max\_binlog\_size = 100M

==> default: -# binlog\_do\_db = include\_database\_name

==> default: -# binlog\_ignore\_db = include\_database\_name

==> default: +bind-address = 0.0.0.0

==> default: [2025-01-18T14:40:43+00:00] INFO: cookbook\_file[/etc/mysql/mysql.conf.d/mysqld.cnf] sending restart action to service[mysql] (immediate)

==> default: \* service[mysql] action restart

==> default: [2025-01-18T14:40:46+00:00] INFO: service[mysql] restarted

==> default:

==> default: - restart service service[mysql]

==> default: \* execute[create\_remote\_user] action run

==> default: [2025-01-18T14:40:46+00:00] INFO: execute[create\_remote\_user] ran successfully

==> default:

==> default: - execute mysql -u root -e "CREATE USER IF NOT EXISTS 'myuser'@'%' IDENTIFIED BY 'mypassword';

==> default: GRANT ALL PRIVILEGES ON \*.\* TO 'myuser'@'%' WITH GRANT OPTION;

==> default: FLUSH PRIVILEGES;"

==> default:

==> default: \* execute[create\_db] action run

==> default: [2025-01-18T14:40:46+00:00] INFO: execute[create\_db] ran successfully

==> default:

==> default: - execute mysql -u root -e "CREATE DATABASE IF NOT EXISTS mydatabase"

==> default: Recipe: jenkins::default

==> default: \* apt\_package[openjdk-11-jdk] action install

==> default: [2025-01-18T14:42:14+00:00] INFO: apt\_package[openjdk-11-jdk] installed openjdk-11-jdk at 11.0.25+9-1ubuntu1~20.04

==> default:

==> default: - install version 11.0.25+9-1ubuntu1~20.04 of package openjdk-11-jdk

==> default: \* apt\_package[net-tools] action install

==> default: [2025-01-18T14:42:18+00:00] INFO: apt\_package[net-tools] installed net-tools at 1.60+git20180626.aebd88e-1ubuntu1

==> default:

==> default: - install version 1.60+git20180626.aebd88e-1ubuntu1 of package net-tools

==> default: \* remote\_file[/tmp/jenkins.deb] action create

==> default: [2025-01-18T14:42:43+00:00] INFO: remote\_file[/tmp/jenkins.deb] created file /tmp/jenkins.deb

==> default:

==> default: - create new file /tmp/jenkins.deb

==> default: [2025-01-18T14:42:45+00:00] INFO: remote\_file[/tmp/jenkins.deb] updated file contents /tmp/jenkins.deb

==> default: - update content in file /tmp/jenkins.deb from none to 5bd95e

==> default: (file sizes exceed 10000000 bytes, diff output suppressed)

==> default:

==> default: \* dpkg\_package[jenkins] action install

==> default: [2025-01-18T14:42:46+00:00] INFO: dpkg\_package[jenkins] installing package(s): jenkins

==> default: [2025-01-18T14:43:33+00:00] INFO: dpkg\_package[jenkins] installed jenkins at 2.387.2

==> default:

==> default: - install version 2.387.2 of package jenkins

==> default:

==> default: \* service[jenkins] action enable

==> default: (up to date)

==> default: \* service[jenkins] action start

==> default: (up to date)

==> default: [2025-01-18T14:43:33+00:00] INFO: Chef Infra Client Run complete in 266.909843985 seconds

==> default: Running handlers:

==> default: [2025-01-18T14:43:33+00:00] INFO: Running report handlers

==> default: Running handlers complete

==> default: [2025-01-18T14:43:33+00:00] INFO: Report handlers complete

==> default: Infra Phase complete, 16/22 resources updated in 04 minutes 28 seconds

If there is an error we can boot the machine with ui mode:

In Vagrant file:

config.vm.provider "virtualbox" **do** |vb|

vb.gui = true

vb.memory = 2048

vb.cpus = 2

end

vagrant halt

vagrant up --provider=virtualbox –debug

User: vagrant

Password: vagrant

Texto

Descripción generada automáticamente

Igual Podemos acceder con el comando

vagrant ssh

PS D:**\U**NIR**\M**aestria**\u**nir**\1**\_semestre**\H**erramientas\_de\_Automatizacion\_de\_Despliegues> vagrant ssh

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-204-generic x86\_64)

\* Documentation: https://help.ubuntu.com

\* Management: https://landscape.canonical.com

\* Support: https://ubuntu.com/pro

System information as of Sat Jan 18 14:25:36 UTC 2025

System load: 1.59

Usage of /: 6.8% of 38.70GB

Memory usage: 28%

Swap usage: 0%

Processes: 133

Users logged in: 0

IPv4 address **for** enp0s3: 10.0.2.15

IPv6 address **for** enp0s3: fd00::a9:1ff:fe47:2532

Expanded Security Maintenance **for** Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.

See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.5 LTS' available.

Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Jan 18 14:25:36 2025 from 10.0.2.2

vagrant@ubuntu-focal:~$

## Comprobación Apache + PHP

Simplemente dentro de nuestro navegador podemos comprobar la página php disponible:

http://localhost:8080/

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

## Comprobación MySQL

Haciendo uso de MySQL Workbench podemos establecer una conexión a nuestra base de datos con el usuario y contraseña definidas en la receta, importante notar el mapeo a otro puerto disponible debido al conflicto de puertos.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Comprobación Jenkins

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

vagrant@ubuntu-focal:/$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword

cc135e0e00da4cfa9572012b629b5149

vagrant@ubuntu-focal:/$

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

# Dificultades

* Instalación del software.
  + Windows: Dentro de la posibles descargas de chef Workstation de manera gráfica con un wizard la página oficial no está actualizada y solo existen versiones antiguas que no tienen todas la funciones que hoy en día se usa, para solucionar este problema fue requerido el uso de chocolatey el cual se instala en powershell y este hace la descarga de la última chef Workstation, eso es importante ya que aunque la máquina de Vagrant se instala con chef solo el problema aparece cuando queremos hacer test ya que la máquina Windows tiene que también tener las capacidades de correr Test Kitchen y para eso es necesario que la instalación sea con una versión reciente ya que el repositorio de código es actualizado a versiones actuales.
  + WSL2: Hay que tener mucha atención a la instalación del software en wsl ya que si bien se pueden instalar todo el software de igual manera pero para la variante Linux, el proveedor Virtual box tiene muchos problemas a la hora de virtualizar debido al directorio compartido entre la máquina host (Windows 11 por ejemplo) y la propia wsl (Debian por ejemplo).
  + Mac: Dentro de mac hay que poner mucha atención a la arquitectura de la computadora, un ejemplo de un problema común es a la hora de utilizar un proveedor de virtualización hay que saber si este proveedor puede soportar nuestra arquitectura, para una computadora Mac pro M2 siendo ARM la arquitectura entonces Virtual Box no es compatible lo cual no nos permitirá virtualizar la máquina definida por Vagrant así que hay alternativas algunas con costo que se tienen que evaluar.
* Limitación de las capacidades.
  + Es posible que las instalaciones proporcionadas por los chef cookbooks estén limitas por las capacidades de la máquina virtual definidas por vagrant, una recomendación es ampliar la memoria y el número de cpus para que las instalaciones se puedan hacer de manera correcta.
* Vagrantfile.
  + El uso de chef como aprovisionador de las instalaciones de la máquina virtual requiere que la licencia sea aceptada, ya que en versiones recientes la versión chef solo no acepta la licencia automáticamente haciendo esto que tengamos que agregar ese paso nosotros mismos.
* Puertos.
  + El mapeo de puertos puede llegar a tener conflictos si los puertos de la máquina host están ya siendo ocupados previamente para ello hacer uso de  auto\_correct: true para poder cambiar a un puerto disponible de manera automática, hay que poner atención a que puerto vagrant decide asignar nuestro servicio.

# Conclusiones

La realización de este proyecto permitió al equipo desarrollar competencias clave en la automatización de despliegues utilizando Chef Workstation. Se logró implementar de manera exitosa la instalación y configuración de servicios como el de Apache con PHP, MySQL y Jenkins, demostrando la efectividad de los cookbooks creados y la correcta gestión de los entornos virtuales a través de Vagrant. A pesar de los desafíos enfrentados, como la instalación de software en entornos heterogéneos y la configuración de puertos, el equipo supo adaptarse y encontrar soluciones adecuadas, fortaleciendo así el trabajo colaborativo y la resolución de problemas.

# Referencias

* Chef. (2023, August 25). About Chef Workstation. About chef workstation. <https://docs.chef.io/workstation/>
* Chocolatey - The package manager for Windows. (n.d.). Chocolatey Software. Retrieved January 18, 2025, from https://chocolatey.org/
* Documentation. (n.d.). Retrieved January 18, 2025, from https://www.ruby-lang.org/en/documentation/
* Chef questions. (n.d.). Chef Questions. Retrieved January 18, 2025, from https://discourse.chef.io/