

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Actividad grupal. Instalación completa de WordPress con Chef y pruebas unitarias y de integración

Integrantes:

- Mikel Álvarez Ruiz de Garibay
- Raúl Félix de la Cruz
- Mohammed Yassine Makhloufi Maleb
- David Recio Arnés

Nota: alvaro.velasco234@comunidadunir.net es integrante del grupo, pero no se ha logrado establecer contacto con él.

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Objetivos

Ejercitar el conocimiento sobre la automatización de las pruebas unitarias y de integración con Chef para validar que la configuración desarrollada. Para llevar a cabo esto, procederemos a la instalación de forma automática de WordPress en una VM (aprovisionada con vagrant) y completada con Chef para posteriormente realizar pruebas unitarias de integración con Chef para validar que la configuración desarrollada cumple con los tests preestablecidos.

Una vez realizado esto, al acceder a la máquina <http://localhost:8081> nos debe mostrar un sencillo blog de WordPress.

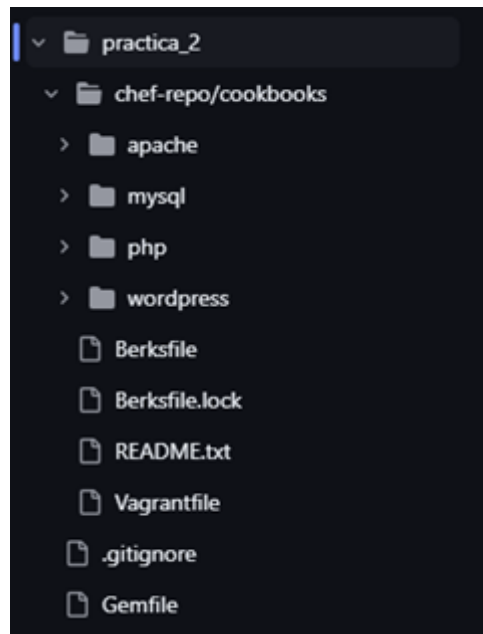
Para la comprobación de las diferentes pruebas realizadas, es necesario ejecutar cada una desde su directorio (receta) o bien ejecución conjunta con:

```
chef exec rspec 'chef-repo/cookbooks/*/spec/*_spec.rb'
```

Estructura del entregable

La estructura se compone de una carpeta general llamada Practica_2, que incluye el archivo Vagrantfile con el que aprovisionar la máquina virtual sobre la que se configurará el servidor web y demás configuraciones para correr WordPress. También se encuentra la carpeta Chef-repo/cookbooks donde se encuentran todas las recetas para la instalación y configuración de los diferentes servicios necesarios para el funcionamiento de WordPress, también los diferentes test de integración y unitarios para el testeo de la instalación.

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025



Aquí se muestra detalle de cómo en el archivo Vagrant se aprovisionan el empleo de cada receta en orden que no altere dependencias.

```

23
24     # Provisionar con Chef Solo después de que Chef esté instalado
25     config.vm.provision :chef_solo do |chef|
26         chef.cookbooks_path = "chef-repo/cookbooks" # Asegúrate de que esta ruta sea correcta
27         chef.add_recipe "apache"
28         chef.add_recipe "mysql"
29         chef.add_recipe "php"
30         chef.add_recipe "wordpress"
31         chef.arguments = "--chef-license accept"
32         chef.install = false
33     end
34 end

```

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Se incluye el archivo Read.me

Este archivo proporciona detalles sobre las credenciales de acceso para WordPress y explica cómo ejecutar pruebas unitarias y de integración para verificar la correcta configuración y funcionamiento de los cookbooks de Chef.

```
admin_user="admin"
admin_password="password"
url: http://localhost:8081/wp-login.php
```

Ejecución de test

Unitarios:

Ejecución de cada una

```
chef exec rspec chef-repo/cookbooks/apache/spec/default_spec.rb
chef exec rspec chef-repo/cookbooks/mysql/spec/default_spec.rb
chef exec rspec chef-repo/cookbooks/php/spec/default_spec.rb
chef exec rspec chef-repo/cookbooks/wordpress/spec/default_spec.rb
```

Ejecución conjunta:

```
chef exec rspec 'chef-repo/cookbooks/*/spec/*_spec.rb'
```

Integración:

en el directorio de wordpress

```
practica_2\chef-repo\cookbooks\wordpress>
```

```
kitchen converge
```

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Aprovisionamiento. (Recetas detalladas)

Destacamos los archivos:

1. Apache recipes/default.db:

```
# Actualiza los repositorios antes de instalar cualquier paquete
apt_update 'Update the apt cache daily' do
  frequency 86400
  action :update
end

# Forzar la instalación de dependencias faltantes antes de continuar
execute 'force-install-dependencies' do
  command 'apt-get install -f'
  user 'root'
  action :run
end

# Instala Apache sin los paquetes recomendados como ssl-cert
package 'apache2' do
  options '--no-install-recommends'
  action :install
  notifies :run, 'execute[force-install-dependencies]', :immediately
end

# Asegúrate de que Apache esté habilitado y en ejecución
service 'apache2' do
  supports status: true, restart: true, reload: true
  action [:enable, :start]
end

# Elimina la configuración predeterminada de Apache
file '/etc/apache2/sites-enabled/000-default.conf' do
  action :delete
  only_if { ::File.exist?('/etc/apache2/sites-enabled/000-default.conf') }
end

# Crea un archivo de configuración virtual para Apache
template '/etc/apache2/sites-available/vagrant.conf' do
  source 'virtual-hosts.conf.erb'
  variables({
    document_root: '/var/www/html' # Define el DocumentRoot para WordPress
  })
  notifies :reload, 'service[apache2]', :immediately
end

# Crea un enlace simbólico para habilitar el sitio
link '/etc/apache2/sites-enabled/vagrant.conf' do
  to '/etc/apache2/sites-available/vagrant.conf'
  notifies :reload, 'service[apache2]', :immediately
end

# Transfiere el archivo index.html a la carpeta raíz de Apache
cookbook_file '/var/www/html/index.html' do
  source 'index.html'
  only_if { ::File.exist?('/etc/apache2/sites-enabled/vagrant.conf') }
end

# Asegúrate de que WordPress se haya colocado en el directorio correcto
execute 'install-wordpress' do
  command 'wget https://wordpress.org/latest.tar.gz -P /var/www/html && tar -xvzf /var/www/html/latest.tar.gz -C /var/www/html && rm /var/www/html/latest.tar.gz'
  user 'root'
  not_if { ::File.exist?('/var/www/html/wordpress') }
  notifies :reload, 'service[apache2]', :immediately
end

# Incluye la receta de facts si es necesaria
include_recipe '::facts'
```

Este archivo realiza lo siguiente:

Actualiza el índice de paquetes de apt diariamente.

Instalación de dependencias: Ejecuta apt-get install -f para corregir dependencias faltantes.

Instalación de Apache: Instala Apache sin paquetes adicionales recomendados.

Habilitar y arrancar Apache: Asegura que Apache esté habilitado y en ejecución.

Eliminar configuración predeterminada: Elimina la configuración del sitio predeterminada de Apache.

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Configurar un host virtual: Crea un archivo de configuración para un host virtual y lo habilita.

Transferencia de index.html: Copia un archivo index.html al directorio raíz de Apache.

Instalación de WordPress: Descarga, descomprime y elimina el archivo de WordPress si no está instalado.

Incluir receta adicional: Incluye una receta facts para más configuraciones o recopilación de información.

2. Mysql recipes/ default.db:

```
# Instalar el servidor de MySQL
package 'mysql-server' do
  action :install
end

# Asegurar que el servicio de MySQL esté corriendo y habilitado
service 'mysql' do
  action [:enable, :start]
  supports status: true, restart: true
end

# Crear la base de datos y el usuario de WordPress si no existen
bash 'create-database' do
  code <<-EOH
    mysql -uroot -e "CREATE DATABASE IF NOT EXISTS wordpress;"
    mysql -uroot -e "CREATE USER IF NOT EXISTS 'wp_user'@'localhost' IDENTIFIED BY 'wp_password';"
    mysql -uroot -e "GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'localhost';"
    mysql -uroot -e "FLUSH PRIVILEGES;"
  EOH
  action :run
end
```

Este script de Chef configura MySQL para un servidor Ubuntu y prepara la base de datos para WordPress. Primero, instala el servidor de MySQL. Luego, asegura que el servicio de MySQL esté habilitado y corriendo. Finalmente, crea la base de datos wordpress y el usuario wp_user con la contraseña wp_password, otorgándole todos los privilegios sobre la base de datos, si no existen ya.

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

3. Php/recipes/ default.db:

```
# Instalar PHP, el módulo de PHP para Apache y MySQL
package 'php' do
  action :install
end

package 'php-mysql' do
  action :install
end

package 'libapache2-mod-php' do
  action :install
end

# Habilita el módulo de PHP y reinicia Apache
execute 'enable-php' do
  command 'systemctl restart apache2'
  action :run
  environment({ 'PATH' => '/bin:/usr/bin' }) # Define el PATH correctamente
  notifies :restart, 'service[apache2]', :immediately
  only_if { ::File.exist?('/etc/apache2/mods-available/php.conf') }
end

# Asegúrate de que Apache esté en ejecución
service 'apache2' do
  supports status: true, restart: true, reload: true
  action [:enable, :start]
end
```

Este script de Chef se encarga de instalar y configurar PHP junto con los módulos necesarios para que funcione con Apache y MySQL.

- Instala PHP:
- Instala el paquete principal de PHP en el sistema.
- Instalación del módulo PHP para MySQL:
- Instala el módulo de PHP que permite la integración con bases de datos MySQL.
- Instalación del módulo de PHP para Apache:
- Instala el módulo que permite que Apache procese scripts PHP.
- Habilitación del módulo PHP y reinicio de Apache
- Reinicia Apache para activar el módulo de PHP, asegurando que el archivo de configuración de PHP esté presente. Se define el PATH para garantizar que los comandos se encuentren correctamente.
- Asegurar que Apache esté habilitado y en ejecución

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

- Habilita Apache para que se inicie automáticamente con el sistema y asegura que esté en ejecución.

4. Wordpress recipes/default.db:

```
# Instalar paquetes necesarios
%w[apache2 php mysql-server].each do |pkg|
  package pkg do
    action :install
  end
end

# Crear directorio de instalación
directory '/var/www/html' do
  owner 'www-data'
  group 'www-data'
  mode '0755'
  action :create
end

# Descargar y extraer WordPress
remote_file '/tmp/wordpress.tar.gz' do
  source 'https://wordpress.org/latest.tar.gz'
  action :create
  not_if { ::File.exist?('/tmp/wordpress.tar.gz') }
end

bash 'extract_wordpress' do
  code <<-EOH
  tar -xzf /tmp/wordpress.tar.gz -C /var/www/html --strip-components=1
  EOH
  creates '/var/www/html/wp-config.php' # Verifica que wp-config.php esté presente
  not_if { ::File.exist?('/var/www/html/wp-config.php') } # Solo extrae si wp-config.php no existe
  action :run
end

# Asegurar que los permisos sean correctos
template '/var/www/html/wp-config.php' do # Archivo wp-config.php en el directorio principal de WordPress
  source 'wp-config.php.erb'
  owner 'www-data'
  group 'www-data'
  mode '0644'
  notifies :restart, 'service[apache2]', :immediately
  action :create
end
```


Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Este script de Chef instala y configura un entorno para WordPress en un servidor Ubuntu. Primero, instala Apache, PHP, y MySQL.

Luego, crea el directorio /var/www/html para WordPress, asegurando los permisos y propietarios adecuados. Descarga y extrae WordPress en ese directorio, creando wp-config.php desde una plantilla y reiniciando Apache si es necesario.

Habilita el sitio de Apache y deshabilita el sitio predeterminado. Cambia la propiedad y los permisos de los archivos de WordPress a www-data. Habilita el módulo de reescritura de Apache y asegura que Apache esté en ejecución.

Instala WP-CLI y lo usa para completar la instalación inicial de WordPress, configurando el sitio y el usuario administrador.

El fichero Berksfile, configurado para hacer referencia a otros cookbooks externos, queda configurado de la siguiente forma:

```

1  source 'https://supermarket.chef.io'
2
3  # Lista de cookbooks requeridos
4  cookbook 'apache2', '~> 8.0'
5  cookbook 'mysql', '~> 10.0'

```

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Criterios de Evaluación

Criterio 1: Instalación y arranque del servidor

Para conseguir que el código se compile adecuadamente y a su vez la instalación, junto con el arranque del servidor web Apache, se ha implementado la receta `apache/default.rb`.

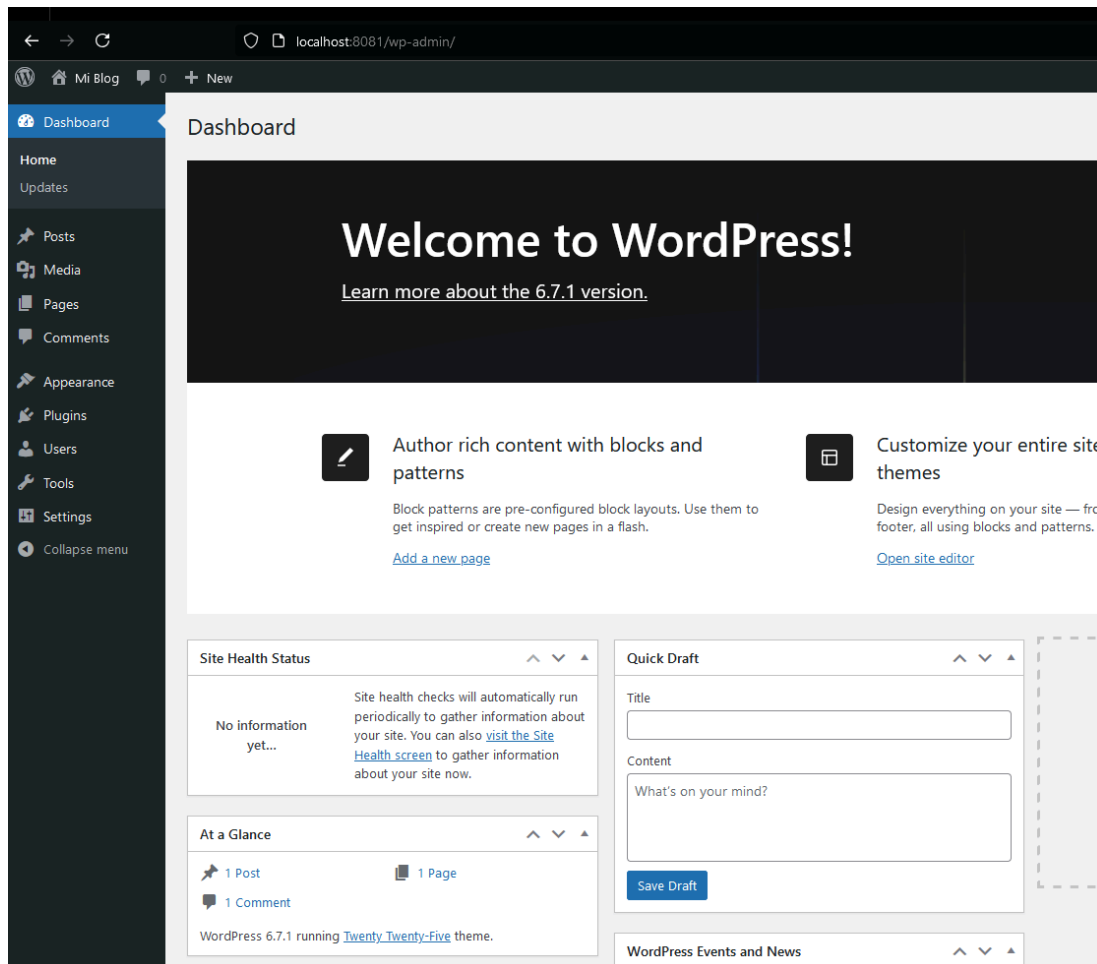
Se instaló correctamente y responde al navegador mostrando la página principal de prueba a través del puerto 8081, pues el 8080 ya se encontraba en uso.

Criterio 2: Instalación de WordPress

Para la instalación de WordPress y su adecuada configuración, se han utilizado 3 recetas adicionales:

- `php`: a través del cual se realiza la instalación de PHP y los módulos necesarios como son el módulo `php-mysql` para la conexión con MySQL, y `libapache2-mod-php` para la integración con Apache. (también reinicia apache tras habilitar el módulo de PHP)
- `mysql`: configura MySQL para WordPress instalando el paquete `mysql-server`, después crea la base de datos llamada WordPress y finalmente crea el usuario `wp_user` con privilegios completos sobre la base de datos.
- `wordpress`: este módulo automatiza la instalación y configuración de WordPress en el servidor mediante chef. Se instalan las dependencias necesarias como `apache`, `php` y `mysql` de forma automática. El sitio por defecto de apache se deshabilita y se configura uno nuevo con soporte para enlaces permanentes. Se descarga y extrae WordPress en `/var/www/html` y a su vez se ajustan los permisos y propiedades para garantizar su correcto funcionamiento. Finalmente, se hace uso de WP-CLI para la inclusión de la conexión a la base de datos, la URL del sitio, y las credenciales del admin.

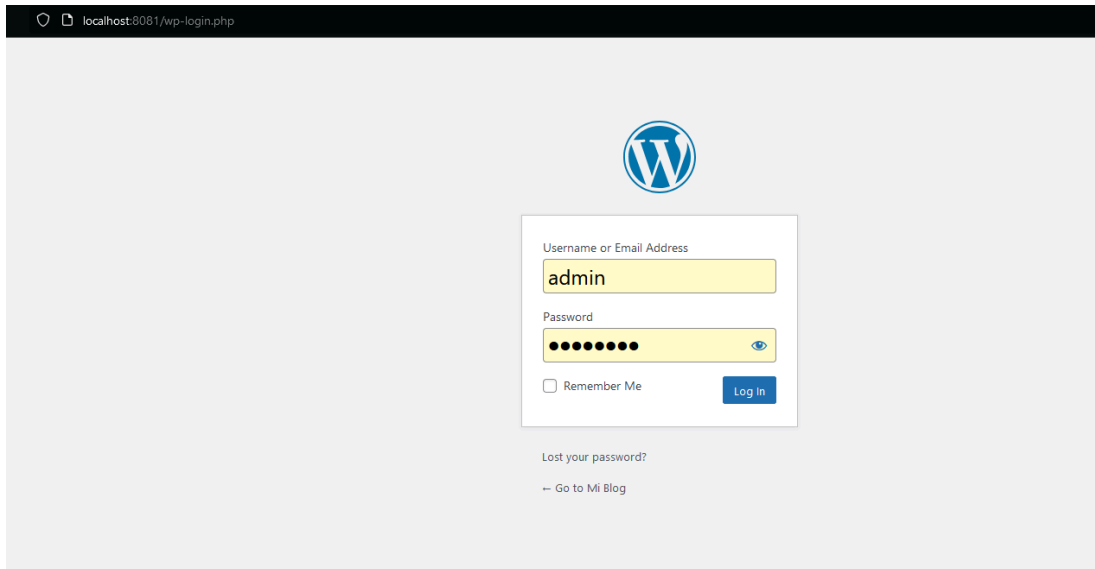
Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025



Criterio 3: Configuración completa de Wordpress

Mediante el uso del módulo `mysql/recipes/default.rb` se automatiza el proceso de restauración de la base de datos de la web en la instancia MySQL de la máquina donde se ha configurado WordPress, asegurando que la web se cargue correctamente al acceder al sitio. En conjunto con el módulo de WordPress, se consigue la conexión a la base de datos y la interacción directa desde la web.

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025



Criterio 4: Pruebas unitarias

Hemos generado una serie de pruebas unitarias 20 en total (hemos desarrollado 5 pruebas unitarias por cada cookbook) donde se comprueba en ambas distribuciones (Ubuntu y CenOS) que se incluyen ciertas recetas, como apache, mysql o php, o la instalación de paquetes como apache o php-mysql.

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Apache /spec/default_spec.db: En la carpeta spec tenemos el código de las pruebas unitarias referentes a la instalación de Apache:

```

1  require 'chefspec'
2
3  describe 'apache::default' do
4    let(:chef_run) do
5      ChefSpec::SoloRunner.new(
6        platform: 'ubuntu',
7        version: '18.04',
8        cookbook_path: ['./chef-repo/cookbooks'] # Ruta correcta a tu directorio de cookbooks
9      ) do |node|
10        # Simular la configuración de la VM según el Vagrantfile
11        node.normal['memory'] = { 'total' => '1024MB' } # Configuración de memoria de Vagrant
12        node.normal['network'] = {
13          interfaces: {
14            'enp0s8' => {
15              addresses: {
16                '192.168.33.40' => { family: 'inet' } # Dirección IP de la red privada en el Vagrantfile
17              }
18            }
19          }
20        }
21        node.normal['apache'] = { 'listen_ports' => ['80'] } # Configuración de puertos de apache
22        end.converge(described_recipe)
23      end
24
25      it 'installs apache2' do
26        expect(chef_run).to install_package('apache2')
27      end
28
29      it 'starts the apache2 service' do
30        expect(chef_run).to start_service('apache2')
31      end
32
33      it 'enables the apache2 service' do
34        expect(chef_run).to enable_service('apache2')
35      end
36    end
  
```

El código verifica que la receta `apache::default` de Chef instale Apache, inicie el servicio, y lo habilite para que se ejecute automáticamente. Las pruebas se realizan en un entorno simulado que imita una máquina virtual con configuraciones específicas de red y memoria.

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Mysql /spec/default_spec.db:

```

1  require 'chefspec'
2
3  describe 'mysql::default' do
4    let(:chef_run) do
5      ChefSpec::SoloRunner.new(
6        platform: 'ubuntu',
7        version: '18.04',
8        cookbook_path: ['./chef-repo/cookbooks'] # Ruta correcta a tu directorio de cookbooks
9      ) do |node|
10        # Aquí puedes definir cualquier configuración adicional del nodo si es necesario.
11        end.converge(described_recipe) # Nombre de la receta a ejecutar
12      end
13    end
14
15    it 'installs the mysql-server package' do
16      expect(chef_run).to install_package('mysql-server')
17    end
18
19    it 'enables the mysql service' do
20      expect(chef_run).to enable_service('mysql')
21    end
22
23    it 'starts the mysql service' do
24      expect(chef_run).to start_service('mysql')
25    end
26  end

```

Este código valida que la receta mysql::default instale el paquete mysql-server, habilite el servicio MySQL para que se inicie automáticamente y también lo inicie. Las pruebas se ejecutan en un entorno simulado de una máquina virtual con Ubuntu 18.04.

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Php /spec/default_spec.db:

```

1  require 'chefspec'
2
3  describe 'php::default' do
4    let(:chef_run) do
5      ChefSpec::SoloRunner.new(
6        platform: 'ubuntu',
7        version: '18.04',
8        cookbook_path: ['./chef-repo/cookbooks']
9      ).converge('php::default')
10   end
11
12   # Verificar que los paquetes estén instalados
13   %w[php php-mysql libapache2-mod-php].each do |pkg|
14     it "installs the #{pkg} package" do
15       expect(chef_run).to install_package(pkg)
16     end
17   end
18
19   # Verificar que Apache se reinicie
20   it 'restarts apache2 service' do
21     execute_resource = chef_run.execute('enable-php')
22     expect(execute_resource).to notify('service[apache2]').to(:restart).immediately
23   end
24
25   # Verificar que el servicio apache2 esté habilitado y en ejecución
26   it 'ensures apache2 service is enabled and running' do
27     expect(chef_run).to enable_service('apache2')
28     expect(chef_run).to start_service('apache2')
29   end
30 end

```

El script anterior verifica que la receta PHP::default instala los paquetes necesarios para PHP, reinicia Apache cuando se habilita PHP, y asegura que el servicio Apache esté habilitado y funcionando correctamente. Las pruebas se ejecutan en un entorno simulado de una máquina virtual con Ubuntu 18.04.

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Mysql /spec/default_spec.db:

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with 'practica_2' at the root, containing 'chef-repo/cookbooks', 'apache', 'mysql', '.vscode', 'compliance', 'recipes', 'spec', and 'test/integration/default'. The 'spec' directory is expanded, showing 'default_spec.rb'. The code editor displays the content of 'default_spec.rb', which is a ChefSpec test file for MySQL. The code includes comments in Spanish and uses the 'ChefSpec::SoloRunner' class to simulate the Chef environment. The test verifies that the 'mysql-server' package is installed, the 'mysql' service is enabled, and the 'mysql' service is started.

```

1  require 'chefspec'
2
3  describe 'mysql::default' do
4    let(:chef_run) do
5      ChefSpec::SoloRunner.new(
6        platform: 'ubuntu',
7        version: '18.04',
8        cookbook_path: ['./chef-repo/cookbooks'] # Ruta correcta a tu directorio de cookbooks
9      ) do |node|
10        # Aquí puedes definir cualquier configuración adicional del nodo si es necesario.
11        end.converge(described_recipe) # Nombre de la receta a ejecutar
12      end
13    end
14
15    it 'installs the mysql-server package' do
16      expect(chef_run).to install_package('mysql-server')
17    end
18
19    it 'enables the mysql service' do
20      expect(chef_run).to enable_service('mysql')
21    end
22
23    it 'starts the mysql service' do
24      expect(chef_run).to start_service('mysql')
25    end
26  end

```

Este código de prueba verifica que la receta mysql::default instala correctamente el paquete MySQL, habilita el servicio MySQL para que se inicie automáticamente, y lo inicia. Las pruebas se ejecutan en un entorno simulado en Ubuntu 18.04.

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Wordpress /spec/default_spec.db:

```

1  require 'chefspec'
2  require 'chefspec/berksheif'
3
4  describe 'wordpress::default' do
5    let(:chef_run) do
6      ChefSpec::SoloRunner.new(
7        platform: 'ubuntu',
8        version: '20.04',
9        cookbook_path: ['./chef-repo/cookbooks'] # Ruta al directorio que contiene el cookbook
10     ).converge('wordpress::default')
11   end
12
13   before do
14     # Stubbing external commands
15     stub_command('apache2ctl -S | grep -q "000-default.conf").and_return(false)
16     stub_command('apache2ctl -S | grep vagrant.conf').and_return(false)
17     stub_command('ls -ld /var/www/html | grep "www-data www-data").and_return(false)
18     stub_command('ls -ld /var/www/html | grep "755").and_return(false)
19     stub_command('apache2ctl -M | grep rewrite_module').and_return(false)
20     stub_command('/usr/local/bin/wp core is-installed --path=/var/www/html').and_return(false)
21   end
22
23   # Instalación de paquetes
24   it 'instala apache2' do
25     expect(chef_run).to install_package('apache2')
26   end
27
28   it 'instala php' do
29     expect(chef_run).to install_package('php')
30   end
31
32   it 'instala mysql-server' do
33     expect(chef_run).to install_package('mysql-server')
34   end
35
36   # Configuración y servicios
37   it 'habilita el servicio apache2' do
38     expect(chef_run).to enable_service('apache2')
39     expect(chef_run).to start_service('apache2')

```

Este código es importante, pues tenemos que manejar las dependencias porque existen algunas pruebas dependen de otros cookbooks.

Por tanto, este código prueba que la receta `wordpress::default` instala y configura correctamente los componentes necesarios para WordPress, incluidos Apache, PHP, MySQL, la creación de directorios y archivos, y la ejecución de scripts necesarios para la configuración inicial.

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Estas pruebas se ejecutan mediante el comando `$ chef exec rspec` y tras ejecutarse, se comprueba que los casos de prueba se han ejecutado satisfactoriamente.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  COMMENTS
PS C:\Users\david\Documents\GitHub\act2HerramientasAuto\practica_2> chef exec rspec 'chef-repo/cookbooks/*/spec/*_spec.rb'
.....

Finished in 2.48 seconds (files took 7.45 seconds to load)
20 examples, 0 failures

```

Criterio 5: Pruebas de integración

A la hora de hacer las pruebas de integración, hemos creado un nuevo archivo llamado `kitchen.yml` en el cual hemos incluido las directrices que chef Kitchen seguirá para hacer las pruebas. En este archivo se especifica el driver que permitirá ejecutar las pruebas (Vagrant para nosotros), el tipo de aprovisionador que usaremos (Chef Zero) y el tipo de verificador (inspec).

```

1  driver:
2    name: vagrant
3
4  provisioner:
5    name: chef_zero
6
7  verifier:
8    name: inspec
9    test_framework: rspec
10
11 platforms:
12   - name: ubuntu-22.04
13     driver:
14       network:
15         - ["forwarded_port", {guest: 80, host: 8081}]
16     verifier:
17       attributes:
18         service: apache2
19         phpmysql: 'php-mysql'
20
21   - name: centos/stream8
22     driver:
23       network:
24         - ["forwarded_port", {guest: 80, host: 8082}]
25     verifier:
26       attributes:
27         service: httpd
28         phpmysql: 'php-mysqlnd'
29
30 suites:
31   - name: default
32     run_list:
33       - recipe[apache::default]
34       - recipe[mysql::default]
35       - recipe[php::default]
36       - recipe[wordpress::default]
37     verifier:
38       inspec_tests:
39         - test/integration/default
40     data_path: "."
41     cookbook_locks: []

```

Por otro lado, en el archivo añadimos también las plataformas en las que queremos hacer las pruebas que en nuestro caso son Ubuntu y CentOS. A estas plataformas les indicamos los puertos que deben usar y las redirecciones respectivas en

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

VirtualBox. Por último especificamos las suites en las que haremos pruebas y la ubicación de los archivos de prueba.

Respecto a las plataformas elegidas, estando ambos sistemas operativos basados en Linux, las similitudes entre ambos son muchas, pero a pesar de eso no hemos sido capaces de hacer que CentOS funcione de la manera que Ubuntu hace. Hemos recibido constantes errores de repositorios inalcanzables desde CentOS que suponemos que originan en la discontinuación del sistema operativo. A pesar de este inconveniente, tras probar con la imagen más reciente de CentOS nos hemos encontrado un error a la hora de que Vagrant descargue la imagen en CentOS 9 y en CentOS Stream 10 recibimos un kernel panic.

```

----> Creating <default-centos-stream10>...
  Bringing machine 'default' up with 'virtualbox' provider...
=> default: Importing base box 'centos/stream10'...
==> default: Matching MAC address for NAT networking...
=> default: Checking if box 'centos/stream10' version '20250106.0' is up to date...
=> default: Setting the name of the VM: kitchen-wordpress-default-centos-stream10-f6788d6e-4065-499f-be6b-bc636fb83d33
=> default: Fixed port collision for 22 => 2222. 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) => 8082 (host) (adapter 1)
  default: 22 (guest) => 2200 (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: 127.0.0.1:2200
  default: SSH username: vagrant
  default: SSH auth method: private key
  default: Warning: Connection reset. Retrying...
The guest machine entered an invalid state while waiting for it
to boot. Valid states are 'starting, running, paused'. The machine is in the
'poweroff' state. Please verify everything is configured
properly and try again.

If the provider you're using has a GUI that comes with it,
it is often helpful to open that and watch the machine, since the
GUI often has more helpful error messages than Vagrant can retrieve.
For example, if you're using VirtualBox, run 'vagrant up' while the
VirtualBox GUI is open.

The primary issue for this error is that the provider you're using
is not properly configured. This is very rarely a Vagrant issue.
>>>>> -----Exception-----
>>>>> Class: Kitchen::ActionFailed
>>>>> Message: 1 actions failed.
>>>>> Failed to complete #create action: [Expected process to exit with [0], but received '1'
---- Begin output of vagrant up --no-provision --provider virtualbox ----
STDOUT: Bringing machine 'default' up with 'virtualbox' provider...
=> default: Importing base box 'centos/stream10'...
=> default: Matching MAC address for NAT networking...
=> default: Checking if box 'centos/stream10' version '20250106.0' is up to date...
=> default: Setting the name of the VM: kitchen-wordpress-default-centos-stream10-f6788d6e-4065-499f-be6b-bc636fb83d33
=> default: Fixed port collision for 22 => 2222. 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) => 8082 (host) (adapter 1)

```

Dicho esto, cabe mencionar que el código está adaptado para detectar correctamente si se trata de un sistema operativo tipo Ubuntu o un CentOS, por lo

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

que no debería haber ningún problema en caso de que funcionaran los repositorios en un futuro.

Una vez creado el fichero, tan solo resta crear las pruebas de integración:

```

1  # Configuración de InSpec
2
3  # Prueba para verificar que el servicio Apache2 esté habilitado y ejecutándose en Ubuntu
4  # En sistemas Ubuntu, el servicio se llama 'apache2'
5  control 'apache-service-ubuntu' do
6    impact 1.0
7    title 'Verificar que Apache2 esté habilitado y en ejecución en Ubuntu'
8    only_if { os[:name] == 'ubuntu' }
9
10   describe service('apache2') do
11     it { should be_enabled } # El servicio debe estar habilitado para iniciarse automáticamente
12     it { should be_running } # El servicio debe estar en ejecución
13   end
14 end
15
16 # Prueba para verificar que el servicio HTTPD esté habilitado y ejecutándose en RedHat/CentOS
17 # En sistemas RedHat/CentOS, el servicio se llama 'httpd'
18 control 'httpd-service-redhat' do
19   impact 1.0
20   title 'Verificar que HTTPD esté habilitado y en ejecución en RedHat/CentOS'
21   only_if { os[:name] == 'redhat' }
22
23   describe service('httpd') do
24     it { should be_enabled } # El servicio debe estar habilitado para iniciarse automáticamente
25     it { should be_running } # El servicio debe estar en ejecución
26   end
27 end
28
29 # Prueba para verificar que el puerto 80 (HTTP) esté escuchando conexiones
30 # Esto asegura que el servidor web esté accesible en el puerto predeterminado
31 control 'port-80-check' do
32   impact 1.0
33   title 'Verificar que el puerto 80 esté escuchando conexiones'
34
35   describe port(80) do
36     it { should be_listening } # El puerto 80 debe estar escuchando
37   end
38 end

```

Entre las pruebas que hemos implementado, encontramos las más típicas que comprueban si los servicios están activados y en marcha, que los puertos importantes, el 80 en este caso, estén abiertos o incluso comprobar si la base de datos de MySQL existe...

```

1  # Configuración de InSpec
2
3  # Prueba para verificar que el servicio MySQL esté habilitado y ejecutándose en Ubuntu
4  # En sistemas Ubuntu, el servicio se llama 'mysql'
5  control 'mysql-service-ubuntu' do
6    impact 1.0
7    title 'Verificar que MySQL esté habilitado y en ejecución en Ubuntu'
8    only_if { os[:name] == 'ubuntu' }
9
10   describe service('mysql') do
11     it { should be_enabled } # El servicio debe estar habilitado para iniciarse automáticamente
12     it { should be_running } # El servicio debe estar en ejecución
13   end
14 end
15
16 # Prueba para verificar que el servicio MariaDB esté habilitado y ejecutándose en RedHat/CentOS
17 # En sistemas RedHat/CentOS, el servicio se llama 'mariadb'
18 control 'mariadb-service-redhat' do
19   impact 1.0
20   title 'Verificar que MariaDB esté habilitado y en ejecución en RedHat/CentOS'
21   only_if { os[:name] == 'redhat' }
22
23   describe service('mariadb') do
24     it { should be_enabled } # El servicio debe estar habilitado para iniciarse automáticamente
25     it { should be_running } # El servicio debe estar en ejecución
26   end
27 end
28
29 # Prueba para verificar que la base de datos de WordPress exista
30 # Se ejecuta un comando SQL para listar las bases de datos y se verifica que 'wordpress' esté en la lista
31 control 'wordpress-database-check' do
32   impact 1.0
33   title 'Verificar que la base de datos WordPress exista'
34
35   describe command('mysql -u root -e "SHOW DATABASES;"') do
36     its('stdout') { should match /wordpress/ } # La salida debe contener la base de datos 'wordpress'
37   end
38 end

```

Por último, ejecutamos “kitchen converge” y esperamos a recibir nuestros resultados por la terminal.

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

```

PS C:\Users\david\Documents\GitHub\act2HerramientasAuto\practica_2\chef-repo\cookbooks\wordpress> kitchen converge
-----> Starting Test Kitchen (v3.5.0)
-----> Creating <default-ubuntu-2204>...
  Bringing machine 'default' up with 'virtualbox' provider...
  ==> default: Importing base box 'bento/ubuntu-22.04'...
==> default: Matching MAC address for NAT networking...
  ==> default: Checking if box 'bento/ubuntu-22.04' version '202407.23.0' is up to date...
  ==> default: Setting the name of the VM: kitchen-wordpress-default-ubuntu-2204-cee6ec9c-3a20-4184-8592-4d76ad213602
  ==> default: Clearing any previously set network interfaces...
  ==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
  ==> default: Forwarding ports...
    default: 80 (guest) => 8881 (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: 7.0.18
    default: VirtualBox Version: 7.1
  ==> default: Setting hostname...
  ==> default: Mounting shared folders...
    default: C:/Users/david/.kitchen/cache => /tmp/omnibus/cache
  ==> default: Machine not provisioned because '--no-provision' is specified.
[SSH] Established

```

```

Preparing files for transfer
Policy lock file doesn't exist, running 'C:\opscode\chef-workstation\bin\chef-cli.BAT install' for Policyfile C:/Users/david/Documents/GitHub/act2HerramientasAuto/practica_2/chef-repo/cookbooks/wordpress/Policyfile.rb

Building policy wordpress
Expanded run list: recipe[apache::default], recipe[mysql::default], recipe[php::default], recipe[wordpress::default]
Caching Cookbooks...
Installing apache >= 0.0.0 from path
Installing mysql >= 0.0.0 from path
Installing php >= 0.0.0 from path
Installing wordpress >= 0.0.0 from path

Lockfile written to C:/Users/david/Documents/GitHub/act2HerramientasAuto/practica_2/chef-repo/cookbooks/wordpress/Policyfile.lock.json
Policy revision id: fd6e736a627c0c4cf563b164ed240a72d1745e54891b176239341e00c8ba8b
Updating policy lock using 'C:\opscode\chef-workstation\bin\chef-cli.BAT update'
Building policy wordpress
Expanded run list: recipe[apache::default], recipe[mysql::default], recipe[php::default], recipe[wordpress::default]
Caching Cookbooks...
Installing apache >= 0.0.0 from path
Installing mysql >= 0.0.0 from path
Installing php >= 0.0.0 from path
Installing wordpress >= 0.0.0 from path

Lockfile written to C:/Users/david/Documents/GitHub/act2HerramientasAuto/practica_2/chef-repo/cookbooks/wordpress/Policyfile.lock.json
Policy revision id: fd6e736a627c0c4cf563b164ed240a72d1745e54891b176239341e00c8ba8b
Preparing dna.json
Exporting cookbook dependencies from Policyfile C:/Users/david/AppData/Local/Temp/default-ubuntu-2204-sandbox-20250109-26052-436tlb using 'C:\opscode\chef-workstation\bin\chef-cli.BAT export' ...
Exported policy 'wordpress' to C:/Users/david/AppData/Local/Temp/default-ubuntu-2204-sandbox-20250109-26052-436tlb

```

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

```
You are installing a package without a version pin. If you are installing
on production servers via an automated process this is DANGEROUS and you will
be upgraded without warning on new releases, even to new major releases.
Letting the version float is only appropriate in test, development or
CI/CD environments.

WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING

Installing chef
Installing with dpkg...
Selecting previously unselected package chef.
(Reading database ... 44902 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-2204>
+-----+
✓ 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-2204 using the validator key.
Using Policyfile 'wordpress' at revision 'fd6e736a627cbc4cf563b164ed240a72d1745e54891b176239341e08c8ba8b'
Resolving cookbooks for run list: ["apache::default@0.1.0 (48015d2)", "mysql::default@0.1.0 (cd97fe1)", "php::default@0.1.0 (b359da7)", "wordpress::default@0.1.0 (3f1ae81)"]
Synchronizing cookbooks:
- php (0.1.0)
- mysql (0.1.0)
- wordpress (0.1.0)
- apache (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 32 resources
```

```
Running handlers:
Running handlers complete
Infra Phase complete, 29/45 resources updated in 02 minutes 07 seconds
Downloading files from <default-ubuntu-2204>
Finished converging <default-ubuntu-2204> (2m50.37s).
```

Asignatura	Grupo	Fecha
Herramientas de Automatización de Despliegues		08/01/2025

Rúbrica

Instalación completa de WordPress con Chef	Descripción	Puntuación máxima (puntos)	Peso %
Criterio 1	El código compila y la ejecución llega al menos a la instalación y arranque del servidor web Apache (al acceder desde un navegador a la máquina virtual, se muestra la página por defecto de Apache)	3	30 %
Criterio 2	La aplicación WordPress está instalada al acceder desde un navegador a la máquina virtual, responde WordPress.	2	20 %
Criterio 3	La aplicación WordPress está completamente configurada: al acceder desde un navegador a la máquina virtual, responde WordPress mostrando directamente el primer blog.	1	10 %
Criterio 4	Se implementan cinco casos de prueba unitaria y al ejecutarlos se demuestra que pasan las pruebas satisfactoriamente.	2	20 %
Criterio 5	Se implementan cinco casos de prueba de integración y al ejecutarlos con Test Kitchen se demuestra que pasan las pruebas satisfactoriamente en un sistema Ubuntu y en un sistema CentOS.	2	20 %