

## **Actividad grupal: Empleo de Chef Workstation**

**Domínguez Urías, Didier**

**Lopez Flamenco, Gerardo**

**Nieto Rivera, Athena**

**Soto Audelo, Jorge**

**Trejo Figueroa, José**

Julio de 2025

## Tabla de contenido

Instalación y configuración de Chef Workstation	3
Actualización del sistema	3
Descarga del instalador de Chef Workstation	3
Instalación del paquete descargado	4
Verificación de la instalación	5
Verificación del binario y estructura interna	5
Generación de estructura base para trabajo con recetas	6
Cookbooks	7
Cookbook 1: fastapi_app	7
Paso 1: Crear el cookbook	7
Paso 2: Crear el archivo de plantilla main.py	8
Paso 3: Escritura detallada de la receta default.rb	9
Paso 4: Ejecución del cookbook y validación del despliegue	12
Cookbook 2	15
Cookbook 3	15
Cookbook 4	15
Cookbook 5	15
Tabla de valoración	16
Reflexión sobre los desafíos enfrentados	16
Conclusiones	17
Recomendaciones	18

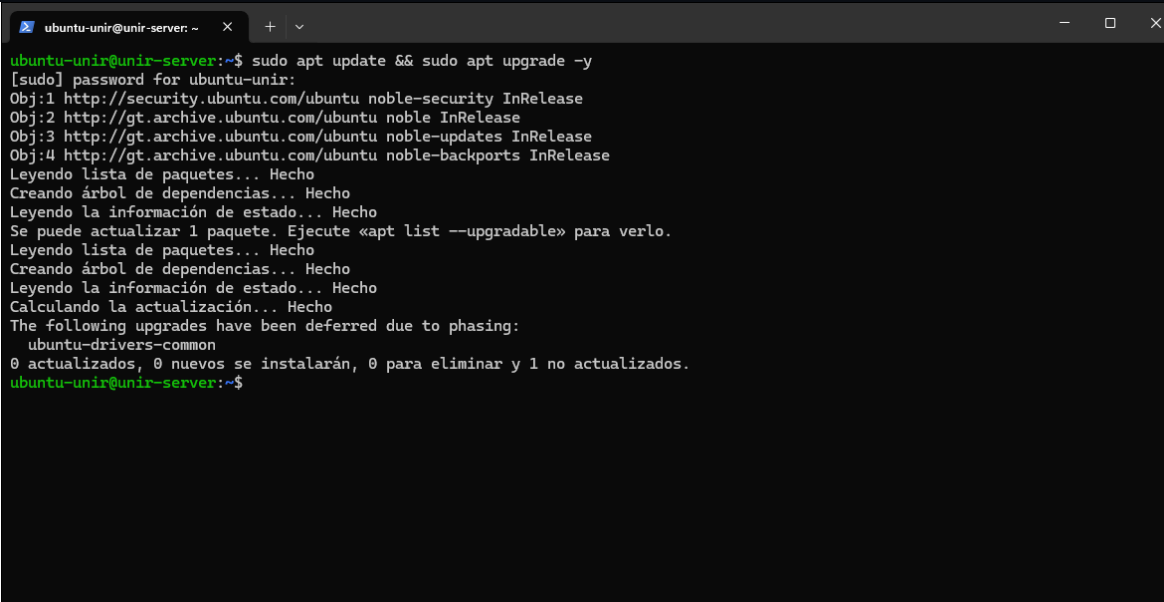
## Instalación y configuración de Chef Workstation

Para comenzar con la actividad, se procedió a realizar la instalación de Chef Workstation en un servidor con sistema operativo Ubuntu Server 24.04 LTS, siguiendo una serie de pasos sistemáticos que aseguran una instalación limpia, actualizada y funcional. A continuación, se detalla el proceso paso a paso:

### Actualización del sistema

Como buena práctica inicial, se actualizó el sistema para garantizar que todos los paquetes estén en su última versión disponible.

```
sudo apt update && sudo apt upgrade -y
```



```
ubuntu-unir@unir-server:~$ sudo apt update && sudo apt upgrade -y
[sudo] password for ubuntu-unir:
Obj:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Obj:2 http://gt.archive.ubuntu.com/ubuntu noble InRelease
Obj:3 http://gt.archive.ubuntu.com/ubuntu noble-updates InRelease
Obj:4 http://gt.archive.ubuntu.com/ubuntu noble-backports InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se puede actualizar 1 paquete. Ejecute «apt list --upgradable» para verlo.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
The following upgrades have been deferred due to phasing:
  ubuntu-drivers-common
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
ubuntu-unir@unir-server:~$
```

### Descarga del instalador de Chef Workstation

Se descargó la versión Chef Workstation 24.12.1073, correspondiente a la LTS más reciente (julio 2025), desde el repositorio oficial de paquetes de Chef.

```
wget https://packages.chef.io/files/stable/chef-
workstation/24.12.1073/ubuntu/20.04/chef-workstation_24.12.1073-1_amd64.deb
```

```
ubuntu-unir@unir-server: ~  
ubuntu-unir@unir-server:~$ wget https://packages.chef.io/files/stable/chef-workstation/24.12.1073/ubuntu/20.04/chef-workstation_24.12.1073-1_amd64.deb  
--2025-07-13 02:27:08-- https://packages.chef.io/files/stable/chef-workstation/24.12.1073/ubuntu/20.04/chef-workstation_24.12.1073-1_amd64.deb  
Resolving packages.chef.io (packages.chef.io)... 146.75.126.110  
Connecting to packages.chef.io (packages.chef.io)[146.75.126.110]:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 225237260 (215M) [application/x-debian-package]  
Saving to: 'chef-workstation_24.12.1073-1_amd64.deb'  
  
chef-workstation_24.12.1073-1_a 100%[=====] 214,80M 20,5MB/s in 1m 48s  
2025-07-13 02:28:57 (1,98 MB/s) - 'chef-workstation_24.12.1073-1_amd64.deb' saved [225237260/225237260]  
  
ubuntu-unir@unir-server:~$ ls -l  
total 219968  
-rw-rw-r-- 1 ubuntu-unir ubuntu-unir 225237260 dic 16 2024 chef-workstation_24.12.1073-1_amd64.deb  
drwxrwxr-x 4 ubuntu-unir ubuntu-unir 4096 jul 5 23:59 web-app  
ubuntu-unir@unir-server:~$  
ubuntu-unir@unir-server:~$ ls -lh  
total 215M  
-rw-rw-r-- 1 ubuntu-unir ubuntu-unir 215M dic 16 2024 chef-workstation_24.12.1073-1_amd64.deb  
drwxrwxr-x 4 ubuntu-unir ubuntu-unir 4,0K jul 5 23:59 web-app  
ubuntu-unir@unir-server:~$
```

## Instalación del paquete descargado

Una vez completada la descarga, se procedió a instalar el paquete **.deb** utilizando **dpkg**.

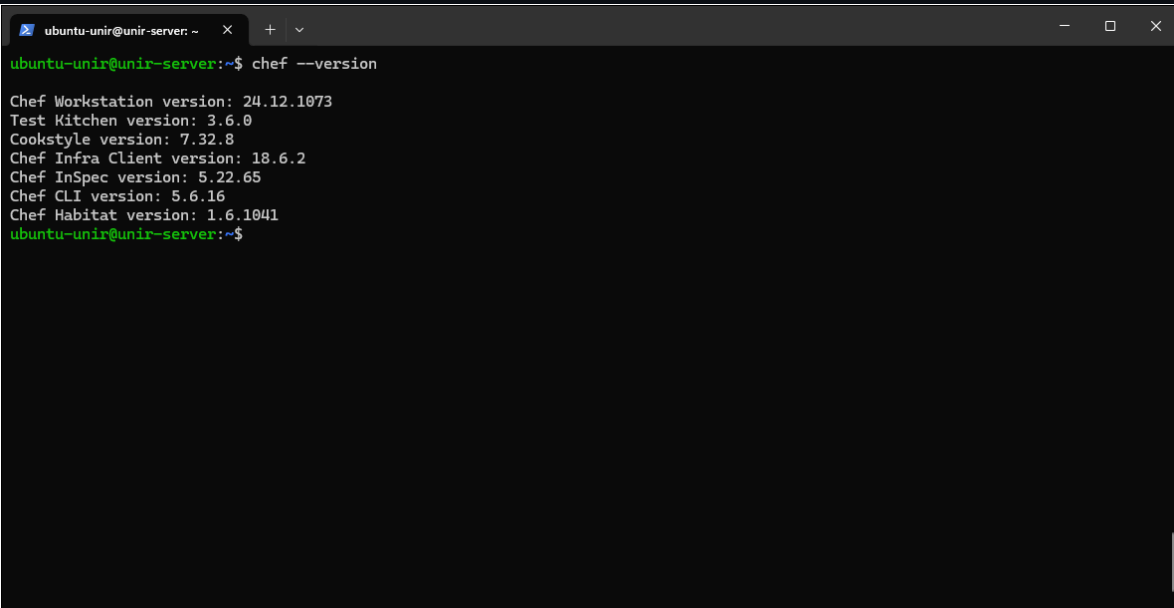
```
sudo dpkg -i chef-workstation_24.12.1073-1_amd64.deb
```

```
ubuntu-unir@unir-server: ~  
ubuntu-unir@unir-server:~$ sudo dpkg -i chef-workstation_24.12.1073-1_amd64.deb  
[sudo] password for ubuntu-unir:  
Seleccionando el paquete chef-workstation previamente no seleccionado.  
(Leyendo la base de datos ... 86975 ficheros o directorios instalados actualmente.)  
Preparando para desempaquetar chef-workstation_24.12.1073-1_amd64.deb ...  
Desempaquetando chef-workstation (24.12.1073-1) ...  
Configurando chef-workstation (24.12.1073-1) ...  
  
Chef Workstation ships with a toolbar application, the Chef Workstation App.  
To run this application some additional dependencies must be installed.  
Using your platform's package manager to install the 'electron' package is  
the easiest way to meet the dependency requirements.  
  
You can then launch the App by running 'chef-workstation-app'.  
The App will then be available in the system tray.  
  
Thank you for installing Chef Workstation!  
You can find some tips on getting started at https://docs.chef.io/workstation/getting\_started/  
  
ubuntu-unir@unir-server:~$ |
```

## Verificación de la instalación

Finalizada la instalación, se verificó que Chef Workstation se hubiera instalado correctamente, mediante el comando:

```
chef --version
```

A terminal window titled 'ubuntu-unir@unir-server: ~' with a dark background. The command 'chef --version' has been entered and executed. The output lists the versions of various Chef components: Chef Workstation (24.12.1073), Test Kitchen (3.6.0), Cookstyle (7.32.8), Chef Infra Client (18.6.2), Chef InSpec (5.22.65), Chef CLI (5.6.16), and Chef Habitat (1.6.1041). The prompt returns to 'ubuntu-unir@unir-server:~\$'.

## Verificación del binario y estructura interna

Para asegurarse de que los ejecutables de Chef estaban correctamente ubicados, se ejecutaron los siguientes comandos:

```
which chef  
ls /opt/chef-workstation/bin/
```

Este último comando listó todos los ejecutables incluidos con la instalación, como *chef-client*, *chef-run*, *knife*, *inspec*, *kitchen*, entre otros.

```
ubuntu-unir@unir-server: ~$ which chef
/usr/bin/chef
ubuntu-unir@unir-server:~$ ls /opt/chef-workstation/bin/
berks      chef-client  chef-solo    delivery    kitchen      stove
chef        chef-resource-inspector  chef-vault   fauxhai     knife
chef-analyze  chef-run    chef-windows-service  foodcritic  mixlib-install
chef-apply   chef-service-manager    chef-zero   hab         ohai
chef-cli     chef-shell  cookstyle    inspec     ruby-env-script.rb
ubuntu-unir@unir-server:~$
```

## Generación de estructura base para trabajo con recetas

Para asegurarse de que los ejecutables de Chef estaban correctamente ubicados, se ejecutaron los siguientes comandos:

```
chef generate repo chef-repo
cd chef-repo
```

Esto creó un directorio **chef-repo** con la estructura base que incluye carpetas para cookbooks, data bags y otras entidades propias del entorno Chef.

```
ubuntu-unir@unir-server: ~/chef-repo
ubuntu-unir@unir-server:~$ chef generate repo chef-repo
Generating Chef Infra repo chef-repo
- Ensuring correct Chef Infra repo file content

Your new Chef Infra repo is ready! Type 'cd chef-repo' to enter it.
ubuntu-unir@unir-server:~$ cd chef-repo/
ubuntu-unir@unir-server:~/chef-repo$ ls
chefignore  cookbooks  data_bags  LICENSE  policyfiles  README.md
ubuntu-unir@unir-server:~/chef-repo$
```

La instalación y configuración de Chef Workstation se realizó de manera exitosa. Todos los binarios fueron correctamente reconocidos, y se estableció el entorno base de trabajo para la posterior creación de recetas y cookbooks. Esta configuración servirá como punto de partida para las siguientes fases de la actividad.

## Cookbooks

### Cookbook 1: fastapi\_app

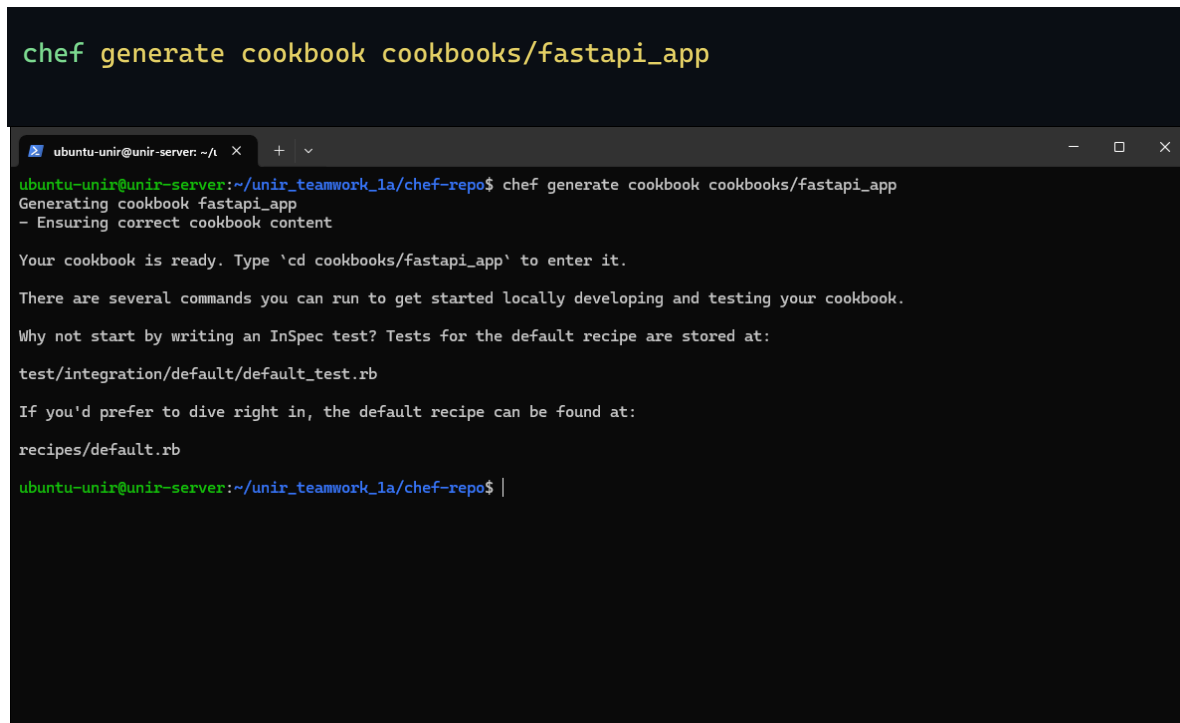
Cookbook personalizado llamado fastapi\_app con el objetivo de automatizar el despliegue de una aplicación web FastAPI sobre un servidor Ubuntu. El proceso incluyó:

- Instalación de herramientas necesarias.
- Creación de un entorno virtual de Python.
- Instalación de dependencias.
- Generación del código fuente (main.py) mediante plantilla.
- Ejecución automatizada de la aplicación mediante un servicio systemd.

#### Paso 1: Crear el cookbook

Desde la terminal, en el directorio *chef-repo*, se ejecutó el siguiente comando:

```
chef generate cookbook cookbooks/fastapi_app
```



```
ubuntu-unir@unir-server: ~/unir_teamwork_1a/chef-repo$ chef generate cookbook cookbooks/fastapi_app
Generating cookbook fastapi_app
- Ensuring correct cookbook content

Your cookbook is ready. Type 'cd cookbooks/fastapi_app' to enter it.

There are several commands you can run to get started locally developing and testing your cookbook.

Why not start by writing an InSpec test? Tests for the default recipe are stored at:
test/integration/default/default_test.rb

If you'd prefer to dive right in, the default recipe can be found at:
recipes/default.rb
ubuntu-unir@unir-server: ~/unir_teamwork_1a/chef-repo$ |
```

Este comando generó la estructura básica del cookbook *fastapi\_app*.

```
ubuntu-unir@unir-server: ~/t  x  +  v
ubuntu-unir@unir-server:~/unir_teamwork_1a/chef-repo$ cd cookbooks/fastapi_app/
ubuntu-unir@unir-server:~/unir_teamwork_1a/chef-repo/cookbooks/fastapi_app$ ls -R
.:
CHANGELOG.md  cheffignore  compliance  kitchen.yml  LICENSE  metadata.rb  Policyfile.rb  README.md  recipes  test
./compliance:
inputs  profiles  README.md  waivers
./compliance/inputs:
./compliance/profiles:
./compliance/waivers:
./recipes:
default.rb
./test:
integration
./test/integration:
default
./test/integration/default:
default_test.rb
ubuntu-unir@unir-server:~/unir_teamwork_1a/chef-repo/cookbooks/fastapi_app$
```

## Paso 2: Crear el archivo de plantilla main.py

Se creó el directorio de plantillas y el archivo:

```
mkdir -p cookbooks/fastapi_app/templates/default
nano cookbooks/fastapi_app/templates/default/main.py.erb
```

Con el siguiente contenido:

```
from fastapi import FastAPI

app = FastAPI(
    title="Demo FastAPI for Chef Workstation",
    description="API de demostración para la tarea del grupo 1_A",
    version="1.0.0"
)

@app.get("/")
def read_root():
    return {
        "message": "Hello from FastAPI"
    }

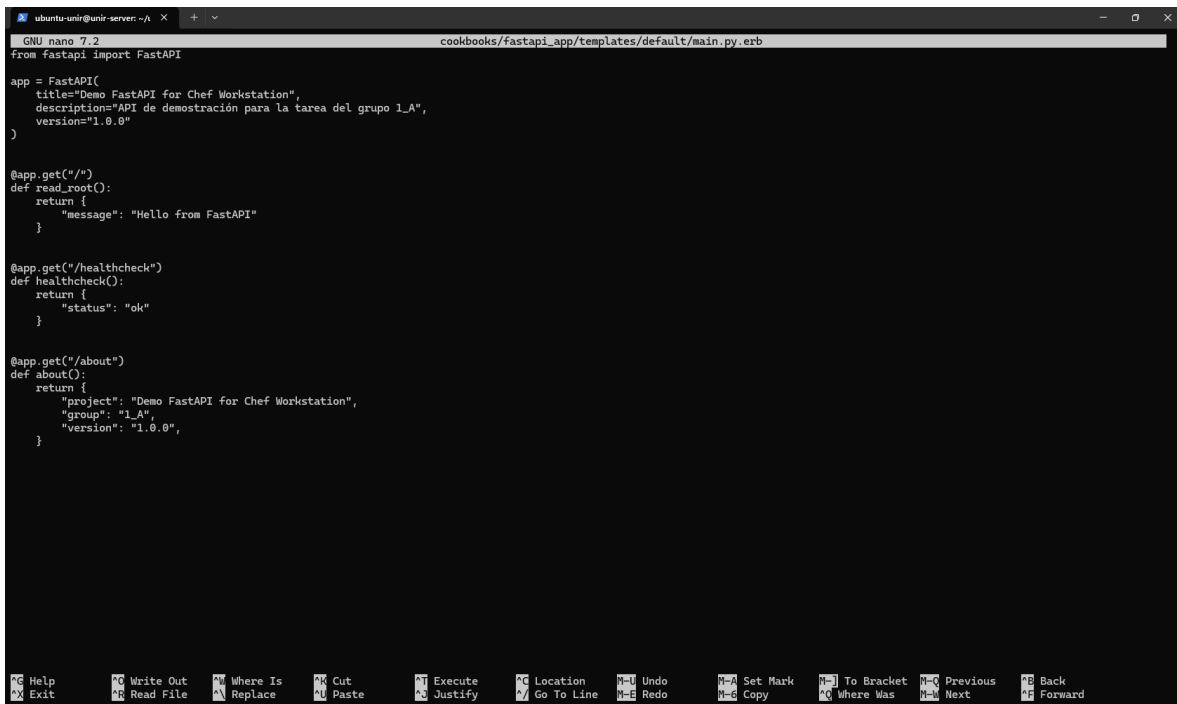
@app.get("/healthcheck")
def healthcheck():
    return {
        "status": "ok"
    }

@app.get("/about")
```



```
def about():
    return {
        "project": "Demo FastAPI for Chef Workstation",
        "group": "1_A",
        "version": "1.0.0",
    }
```

Con el siguiente contenido:



```
GNU nano 7.2 cookbooks/fastapi_app/templates/default/main.py.erb
from fastapi import FastAPI

app = FastAPI(
    title="Demo FastAPI for Chef Workstation",
    description="API de demostración para la tarea del grupo 1_A",
    version="1.0.0"
)

@app.get("/")
def read_root():
    return {
        "message": "Hello from FastAPI"
    }

@app.get("/healthcheck")
def healthcheck():
    return {
        "status": "ok"
    }

@app.get("/about")
def about():
    return {
        "project": "Demo FastAPI for Chef Workstation",
        "group": "1_A",
        "version": "1.0.0",
    }
```

### Paso 3: Escritura detallada de la receta default.rb

Con el cookbook *fastapi\_app* ya generado y la plantilla *main.py.erb* lista, se procedió a construir la receta principal *default.rb*, ubicada en:

```
cookbooks/fastapi_app/recipes/default.rb
```

Esta receta fue organizada en bloques funcionales claramente definidos, utilizando recursos nativos de Chef para asegurar una implementación idempotente, ordenada y reutilizable.

### Actualizar el sistema y preparar el entorno

Se actualiza el sistema operativo para evitar conflictos con versiones obsoletas de paquetes y se instalan las herramientas necesarias:

```
execute 'apt_update' do
    command 'apt update && apt upgrade -y'
```

```

end

package %W(python3-venv curl) do
  action :install
end

```

### Crear el directorio del proyecto

```

directory '/home/ubuntu-unir/fastapi_app' do
  owner 'ubuntu-unir'
  group 'ubuntu-unir'
  mode '0755'
  recursive true
  action :create
end

```

Este directorio contendrá todo el proyecto: entorno virtual, archivo de aplicación y scripts.

### Crear entorno virtual

```

execute 'crear entorno virtual' do
  command 'python3 -m venv .venv'
  cwd '/home/ubuntu-unir/fastapi_app'
  user 'ubuntu-unir'
  environment({ 'HOME' => '/home/ubuntu-unir' })
  not_if {
    ::File.exist?('/home/ubuntu-unir/fastapi_app/.venv/bin/activate')
  }
end

```

Se crea el entorno solo si no existe, y se ejecuta como el usuario **ubuntu-unir** para evitar errores de permisos.

### Instalar FastAPI

```

execute 'instalar fastapi' do
  command '/home/ubuntu-unir/fastapi_app/.venv/bin/pip install "fastapi[standard]"'
  cwd '/home/ubuntu-unir/fastapi_app'
  user 'ubuntu-unir'
  environment({
    'HOME' => '/home/ubuntu-unir',
    'PATH' => '/home/ubuntu-unir/fastapi_app/.venv/bin:/usr/bin:/bin'
  })
end

```

Se instalan las dependencias en el entorno virtual, asegurando aislamiento del sistema principal

### Generar main.py desde plantilla

```
template '/home/ubuntu-unir/fastapi_app/main.py' do
  source 'main.py.erb'
  owner 'ubuntu-unir'
  group 'ubuntu-unir'
  mode '0644'
end
```

Este archivo define los endpoints `/`, `/healthcheck` y `/about` de la aplicación FastAPI.

### Crear servicio systemd fastapi.service

```
file '/etc/systemd/system/fastapi.service' do
  content <<~UNIT
    [Unit]
    Description=FastAPI Application
    After=network.target

    [Service]
    User=ubuntu-unir
    WorkingDirectory=/home/ubuntu-unir/fastapi_app
    ExecStart=/home/ubuntu-unir/fastapi_app/.venv/bin/python -m fastapi
run main.py --host 0.0.0.0 --port 8080
    Restart=always

    [Install]
    WantedBy=multi-user.target
  UNIT
  mode '0644'
end
```

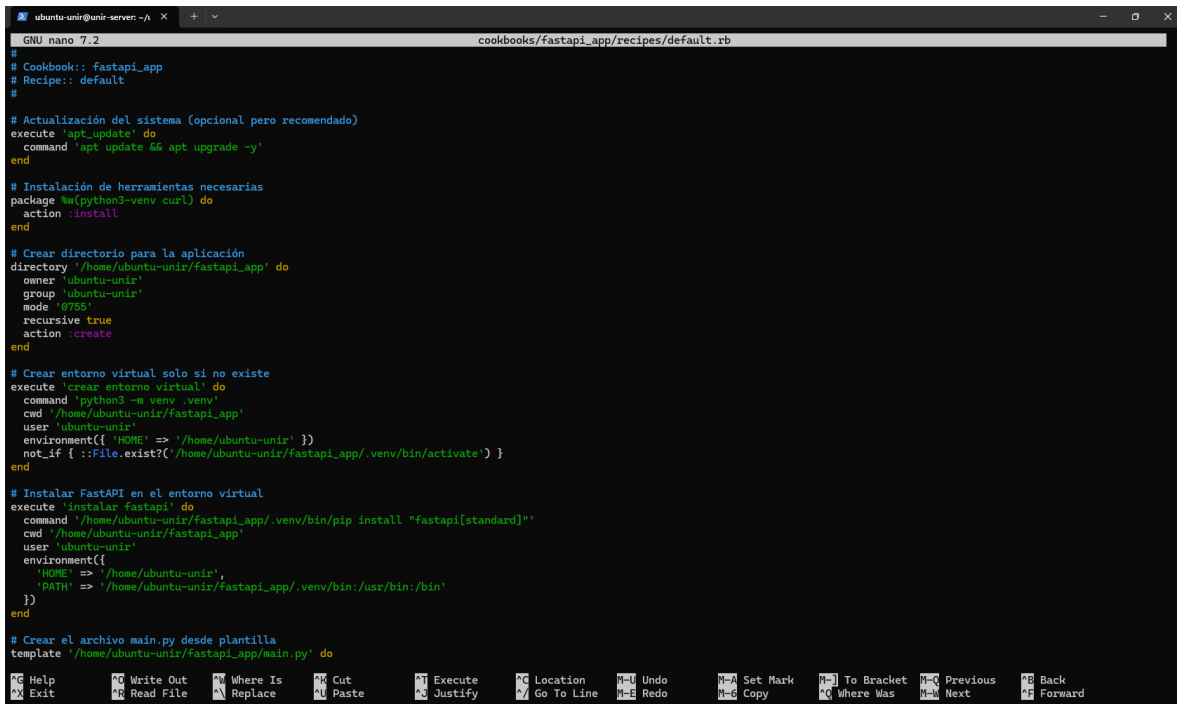
Este archivo define cómo se ejecutará la aplicación como servicio del sistema, de forma persistente y con reinicio automático en caso de falla.

### Recargar systemd y activar el servicio

```
execute 'reload systemd' do
  command 'systemctl daemon-reexec && systemctl daemon-reload'
end

service 'fastapi' do
  action [:enable, :start]
end
```

Con esto, el servicio **fastapi** queda habilitado para iniciarse automáticamente con el sistema y se activa de inmediato.



```
GNU nano 7.2 cookbooks/fastapi_app/recipes/default.rb
#
# Cookbook:: fastapi_app
# Recipe:: default
#
# Actualización del sistema (opcional pero recomendado)
execute 'apt update' do
  command 'apt update && apt upgrade -y'
end

# Instalación de herramientas necesarias
package %([python3-venv curl]) do
  action :install
end

# Crear directorio para la aplicación
directory '/home/ubuntu-unir/fastapi_app' do
  owner 'ubuntu-unir'
  group 'ubuntu-unir'
  mode '0755'
  recursive true
  action :create
end

# Crear entorno virtual solo si no existe
execute 'crear entorno virtual' do
  command 'python3 -m venv .venv'
  cwd '/home/ubuntu-unir/fastapi_app'
  user 'ubuntu-unir'
  environment({'HOME' => '/home/ubuntu-unir'})
  not_if { ::File.exist?('/home/ubuntu-unir/fastapi_app/.venv/bin/activate') }
end

# Instalar FastAPI en el entorno virtual
execute 'instalar fastapi' do
  command '/home/ubuntu-unir/fastapi_app/.venv/bin/pip install "fastapi[standard]"'
  cwd '/home/ubuntu-unir/fastapi_app'
  user 'ubuntu-unir'
  environment({'HOME' => '/home/ubuntu-unir',
               'PATH' => '/home/ubuntu-unir/fastapi_app/.venv/bin:/usr/bin:/bin'})
end

# Crear el archivo main.py desde plantilla
template '/home/ubuntu-unir/fastapi_app/main.py' do
```

## Paso 4: Ejecución del cookbook y validación del despliegue

Una vez completada la receta **default.rb**, se procedió a su ejecución para desplegar la aplicación FastAPI de forma automatizada.

### Ejecutar el cookbook con Chef

```
sudo chef-client --local-mode --runlist 'recipe[fastapi_app]'
```

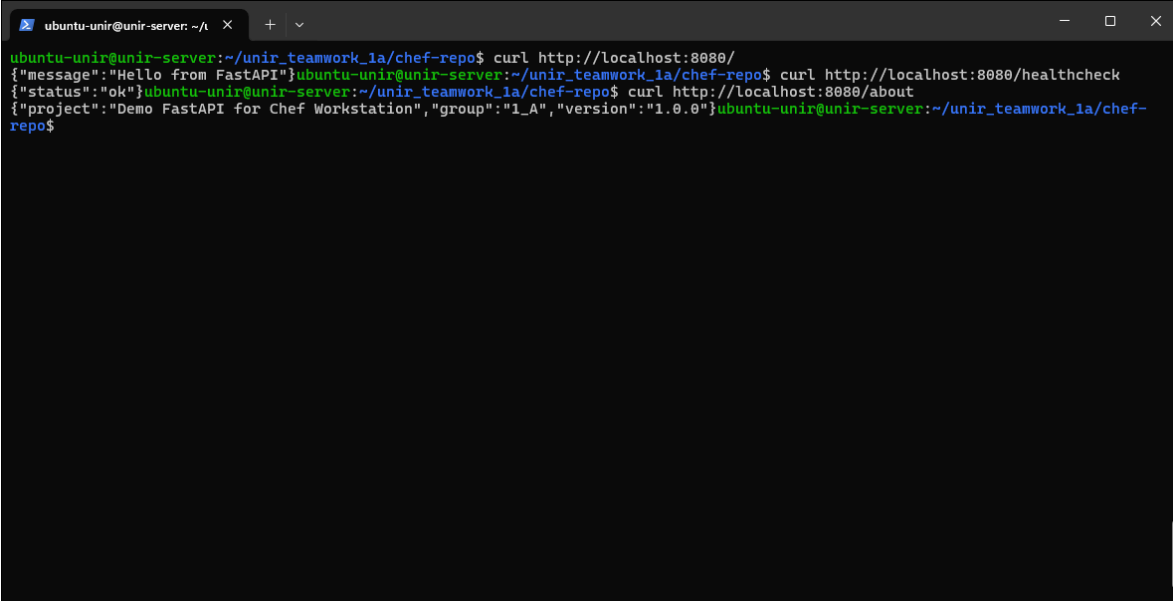
Este comando indica a Chef que ejecute en modo local (**--local-mode**) y aplique específicamente la receta **fastapi\_app**.



## Validar acceso a la API

Se realizaron peticiones HTTP locales para confirmar que la aplicación FastAPI se encontraba en funcionamiento y respondiendo correctamente a sus rutas definidas.

```
curl http://localhost:8080/
curl http://localhost:8080/healthcheck
curl http://localhost:8080/about
```



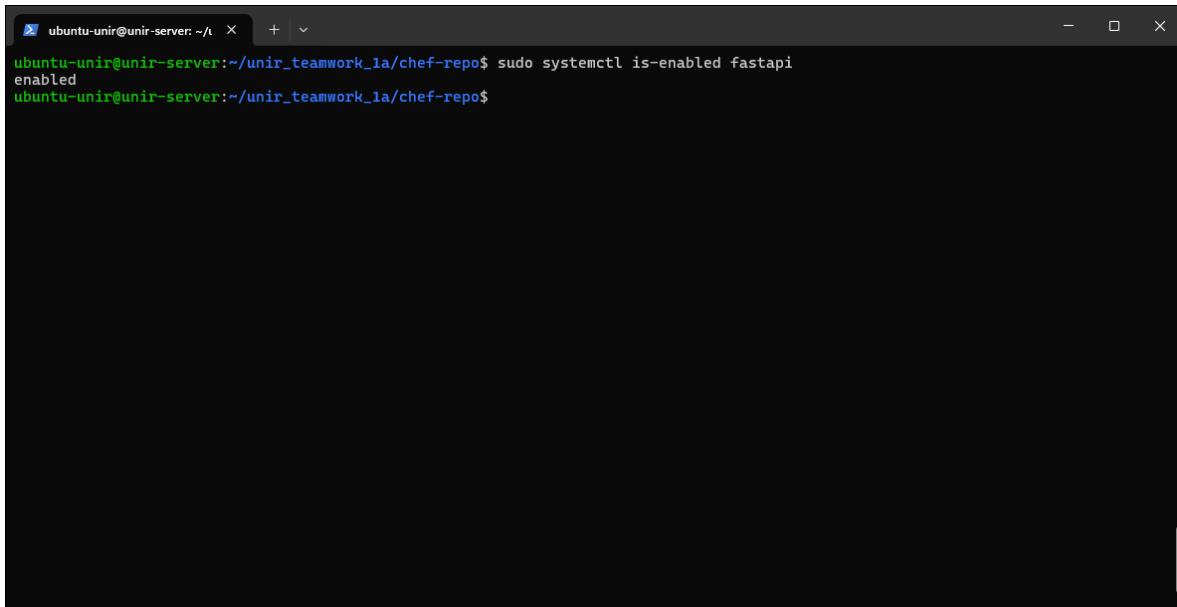
The image shows a terminal window with the following content:

```
ubuntu-unir@unir-server: ~/unir_teamwork_1a/chef-repo$ curl http://localhost:8080/
{"message": "Hello from FastAPI"}ubuntu-unir@unir-server: ~/unir_teamwork_1a/chef-repo$ curl http://localhost:8080/healthcheck
{"status": "ok"}ubuntu-unir@unir-server: ~/unir_teamwork_1a/chef-repo$ curl http://localhost:8080/about
{"project": "Demo FastAPI for Chef Workstation", "group": "1_A", "version": "1.0.0"}ubuntu-unir@unir-server: ~/unir_teamwork_1a/chef-repo$
```

## Verificar inicio automático del servicio

Para confirmar que el servicio se inicia automáticamente con el sistema, se ejecutó:

```
sudo systemctl is-enabled fastapi
```

A terminal window with a dark background. The prompt is 'ubuntu-unir@unir-server: ~/unir\_teamwork\_1a/chef-repo\$'. The command 'sudo systemctl is-enabled fastapi' has been entered, and the output 'enabled' is displayed on the next line. The prompt is now 'ubuntu-unir@unir-server: ~/unir\_teamwork\_1a/chef-repo\$'.

El despliegue fue exitoso. La aplicación FastAPI quedó:

- Ejecutándose en el puerto 8080.
- Expuesta como servicio systemd.
- Respondiendo a todas las rutas esperadas.
- Preparada para iniciar automáticamente al reiniciar el servidor.

Esto demuestra que la receta es funcional, reproducible y adecuada para producción o entornos educativos.

## Cookbook 2: RunSpringbootAPP

Cookbook personalizado llamado RunSpringbootApp con el objetivo de automatizar el despliegue de una aplicación web Java desde un repositorio remoto en un ambiente de MacOS. El proceso incluye:

- Definición de variables importantes.
- Instalación de dependencias usando brew.
- Clonado de un repositorio público.
- Construcción del JAR.
- Eliminar un proceso ya existente de java en caso de que lo haya.
- Ejecución automatizada de la aplicación mediante Java.

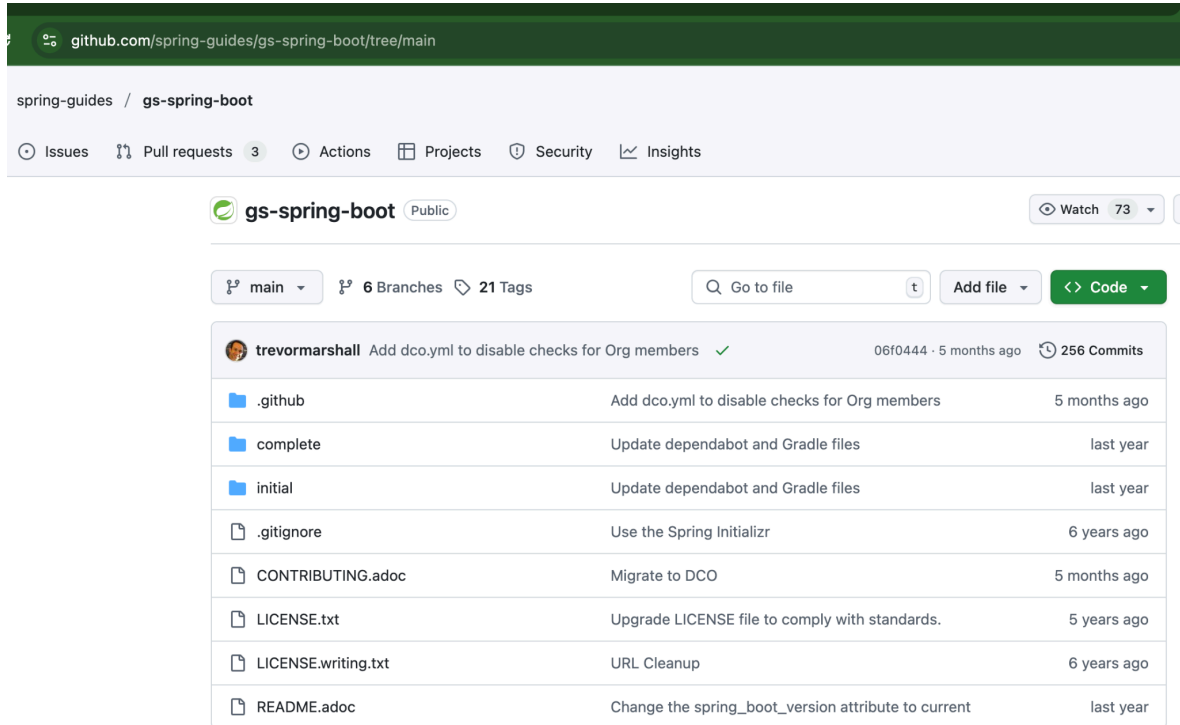
Se ejecutò el siguiente comando para generar el cookbook

```
chef generate cookbook cookbooks/RunSpringbootAppCookbook 3
```

## Paso 1: Se seleccionò el repositorio:

<https://github.com/spring-guides/gs-spring-boot/tree/main>

Como ejemplo para el cookbook, se utilizarà uno de los repositorios ejemplos oficiales de springboot, el cual simplemente inicia un servidor web en el puerto 8080, al enviar peticiones GET a la raiz del servidor, este te responde con "Greetings from Spring Boot!".



spring-guides / gs-spring-boot

Issues Pull requests 3 Actions Projects Security Insights

gs-spring-boot Public Watch 73

main 6 Branches 21 Tags Go to file Add file Code

trevormarshall	Add dco.yml to disable checks for Org members ✓	06f0444 · 5 months ago	256 Commits
.github	Add dco.yml to disable checks for Org members	5 months ago	
complete	Update dependabot and Gradle files	last year	
initial	Update dependabot and Gradle files	last year	
.gitignore	Use the Spring Initializr	6 years ago	
CONTRIBUTING.adoc	Migrate to DCO	5 months ago	
LICENSE.txt	Upgrade LICENSE file to comply with standards.	5 years ago	
LICENSE.writing.txt	URL Cleanup	6 years ago	
README.adoc	Change the spring_boot_version attribute to current	last year	



## Paso 2: Se crea la receta en default.rb:

```
default.rb M X
chef-repo > cookbooks > runSpringbootAPP > recipes > default.rb
You, 24 minutes ago | 1 author (You)
1 #Receta que instala dependencias (Solo MacOS), clona un repositorio, construye un jar y corre un aplicacion de java
2 #Se define el url del repositorio publico de github
3 REPO_URL = "https://github.com/spring-guides/gs-spring-boot.git"
4 #Se define como se llamara la app de springboot
5 APP_NAME = "your-springboot-api"
6 #Se define el directorio donde se clonara el repo
7 APP_DIR = "#{ENV['HOME']}/#{APP_NAME}"
8 #Se define el sub-directorio donde se encuentra el archivo pom.xml dentro del repositorio (en el caso del repositorio de ejemplo, se encuentra en /complete/)
9 APP_SUBDIR = "#{APP_DIR}/complete"
10
11
12 #Instalamos las dependencias usando brew
13 homebrew_package 'git' do
14   action :install
15 end
16 homebrew_package 'openjdk' do
17   action :install
18 end
19 homebrew_package 'maven' do
20   action :install
21 end
22
23
24 # Clonamos el repositorio de la rama main
25 git APP_DIR do
26   repository REPO_URL
27   revision 'main'
28   action :sync
29 end
30
31 # Construimos el JAR usando maven
32 execute 'build spring boot app' do
33   cwd APP_SUBDIR
34   command 'mvn clean package -DskipTests'
35 end
36
37 #Eliminamos un proceso de java corriendo en caso de que exista uno
38 execute 'kill existing spring boot app' do
39   command "pkill -f 'java -jar' || true"
40   only_if "pgrep -f 'java -jar'"
41 end
42
43 # Corremos la aplicacion de java
44 execute 'run spring boot app' do
45   cwd APP_SUBDIR
46   command "nohup java -jar target/*.jar"
47 end
You, 24 minutes ago • Feat: Se agrega cookbook RunSpringbootAPP ...
```

## Paso 3: Se definen las variables generales (su proposito viene comentado dentro de la receta)

```
3 #Se define el url del repositorio publico de github
4 REPO_URL = "https://github.com/spring-guides/gs-spring-boot.git"
5 #Se define como se llamara la app de springboot
6 APP_NAME = "your-springboot-api"
7 #Se define el directorio donde se clonara el repo
8 APP_DIR = "#{ENV['HOME']}/#{APP_NAME}"
9 #Se define el sub-directorio donde se encuentra el archivo pom.xml dentro del repositorio (en el caso del repositorio de ejemplo, se encuentra en /complete/)
10 APP_SUBDIR = "#{APP_DIR}/complete"
11
```

## Paso 4: Se ejecuta el cookbook con el siguiente comando

```
sudo chef-client --local-mode --runlist 'recipe[RunSpringbootAPP]'
```

```

coppel@192 chef-repo % sudo chef-client --local-mode --runlist 'recipe[runSpringbootAP]'

[2025-07-13T16:54:03-07:00] WARN: No config file found or specified on command line. Using command line options instead.
Chef Infra Client, version 18.7.10
Patents: https://www.chef.io/patents
Infra Phase starting
Resolving cookbooks for run list: ["runSpringbootAP"]
Synchronizing cookbooks:
  - runSpringbootAP (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 7 resources
Recipe: runSpringbootAP::default
  * homebrew_package[git] action install (up to date)
  * homebrew_package[openjdk] action install (up to date)
  * homebrew_package[maven] action install (up to date)
  * git[/Users/coppel/your-springboot-ap] action sync (up to date)
  * execute[build spring boot app] action run
    - execute mvn clean package -DskipTests
  * execute[kill existing spring boot app] action run
    - execute pkill -f 'java -jar' || true
  * execute[run spring boot app] action run^C[2025-07-13T17:00:12-07:00] FATAL: SIGINT received, stopping

^C[2025-07-13T17:00:15-07:00] FATAL: SIGINT received, stopping

Running handlers:
[2025-07-13T17:00:15-07:00] ERROR: Running exception handlers
Running handlers complete
[2025-07-13T17:00:15-07:00] ERROR: Exception handlers complete
Infra Phase failed. 2 resources updated in 06 minutes 11 seconds
[2025-07-13T17:00:15-07:00] FATAL: Stacktrace dumped to /Users/coppel/.chef/local-mode-cache/cache/chef-stacktrace.out
[2025-07-13T17:00:15-07:00] FATAL:
[2025-07-13T17:00:15-07:00] FATAL: PLEASE PROVIDE THE CONTENTS OF THE stacktrace.out FILE (above) IF YOU FILE A BUG REPORT
[2025-07-13T17:00:15-07:00] FATAL:
coppel@192 chef-repo % sudo chef-client --local-mode --runlist 'recipe[runSpringbootAP]'

Password:
[2025-07-13T17:00:28-07:00] WARN: No config file found or specified on command line. Using command line options instead.
Chef Infra Client, version 18.7.10
Patents: https://www.chef.io/patents
Infra Phase starting
Resolving cookbooks for run list: ["runSpringbootAP"]
Synchronizing cookbooks:
  - runSpringbootAP (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 7 resources
Recipe: runSpringbootAP::default
  * homebrew_package[git] action install (up to date)
  * homebrew_package[openjdk] action install (up to date)
  * homebrew_package[maven] action install (up to date)
  * git[/Users/coppel/your-springboot-ap] action sync (up to date)
  * execute[build spring boot app] action run
    - execute mvn clean package -DskipTests
  * execute[kill existing spring boot app] action run
    - execute pkill -f 'java -jar' || true
  * execute[run spring boot app] action run

```

**Paso 5: Se accede al navegador al enlace: localhost:8080 para verificar el correcto funcionamiento del aplicativo**



**Greetings from Spring Boot!**

**Cookbook 4**

**Cookbook 5**

## Tabla de valoración

	Sí	No	A veces
Todos los miembros se han integrado al trabajo del grupo			
Todos los miembros participan activamente			
Todos los miembros respetan otras ideas aportadas			
Todos los miembros participan en la elaboración del informe			
Me he preocupado por realizar un trabajo cooperativo con mis compañeros			
Señala si consideras que algún aspecto del trabajo en grupo no ha sido adecuado			

## Reflexión sobre los desafíos enfrentados

Durante el desarrollo de la actividad, el equipo enfrentó varios desafíos técnicos y organizativos que requirieron investigación, comunicación y toma de decisiones colaborativa. A continuación, se enumeran los principales obstáculos encontrados:

- **Configuración inicial de Chef Workstation en Ubuntu Server:** Uno de los principales obstáculos fue la configuración correcta del entorno de Chef Workstation en Ubuntu Server, especialmente en lo relativo a la disponibilidad del comando chef, las rutas del binario y la necesidad de configurar correctamente el entorno de shell para evitar errores como command not found.
- **Problemas al intentar usar nohup desde el cookbook 1 (fastapi\_app):** Se descubrió que Chef no puede gestionar correctamente procesos que quedan en segundo plano con nohup, ya que este comando bloqueaba la ejecución de la receta. Se reemplazó por una solución más robusta con la creación de un servicio systemd, que permite iniciar, detener y monitorear la aplicación de forma segura.
- **Idempotencia de la receta Chef:** Algunos comandos se debieron condicionar con not\_if para evitar reejecuciones innecesarias y garantizar que la receta fuera idempotente (repetible sin efectos adversos).
- **Separación entre lógica de configuración y contenido dinámico:** Se aprendió a utilizar plantillas (template) para generar archivos como main.py, manteniendo la receta más clara y reutilizable, separando la configuración de los contenidos específicos de la aplicación.

## Conclusiones

- La herramienta Chef Workstation demostró ser poderosa y flexible para la automatización de despliegues, siempre que se estructure correctamente la lógica en cookbooks bien definidos.
- Es posible automatizar no solo la instalación de paquetes, sino también el despliegue completo de aplicaciones modernas, incluyendo su ejecución como servicios del sistema.
- El enfoque modular y declarativo de Chef favorece la reproducibilidad, una característica clave para entornos DevOps, pruebas e infraestructura como código.
- La experiencia permitió al equipo desarrollar habilidades técnicas prácticas, trabajar en equipo y documentar un proceso técnico completo.

## Recomendaciones

- Para futuros equipos, se recomienda probar primero todos los pasos manualmente, antes de codificarlos en una receta. Esto ayuda a entender mejor los comandos, rutas y posibles errores.
- Evitar el uso de comandos como sudo dentro de las recetas Chef, ya que Chef se ejecuta como root.
- Siempre validar rutas, usuarios y permisos cuando se utilizan servicios como systemd.
- Aprovechar los recursos oficiales de documentación de Chef y los foros de la comunidad cuando surgen problemas específicos.
- Tomar capturas de pantalla durante cada etapa para no olvidar pasos importantes y facilitar la documentación.