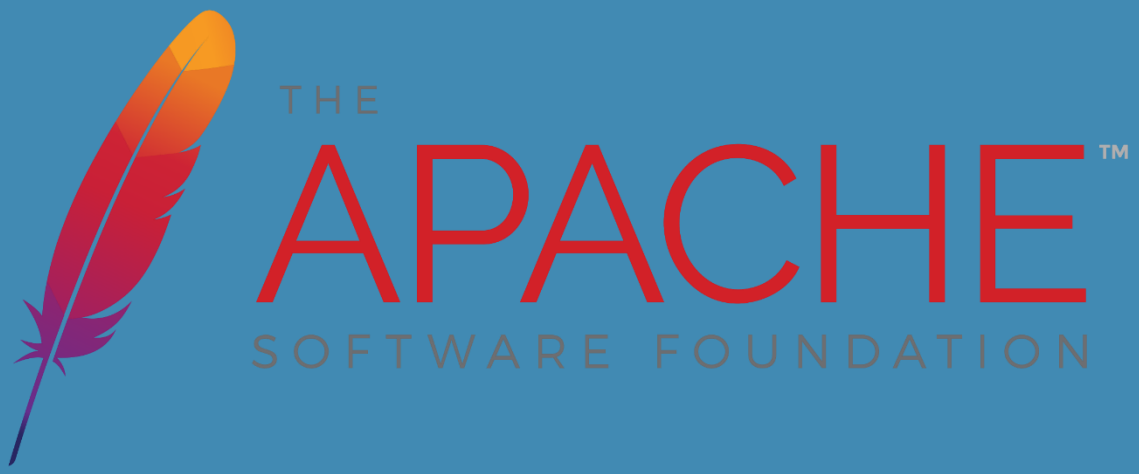
INTRODUCCIÓN APLICACIONES WEB CON APACHE



Eric Serrano Marín

CETI Puesta en Producción Segura

ÍNDICE

1. INTRODUCCIÓN.

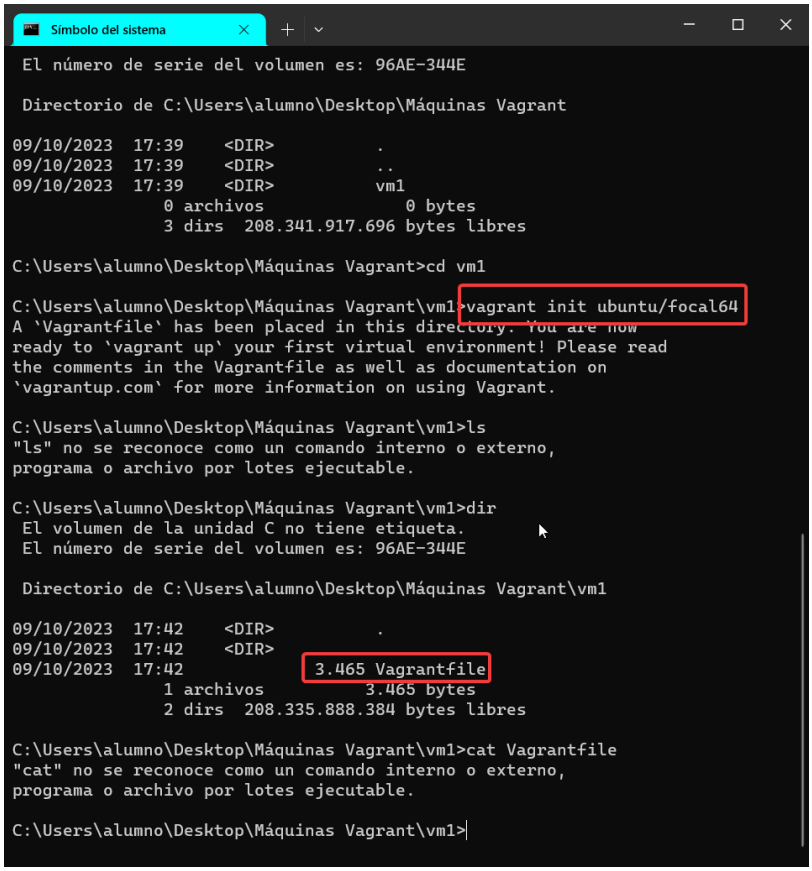
Apache es uno de los servidores web con más tradición en el mundo. Por defecto ofrece contenido estático, pero también es capaz de ofrecer contenido dinámico con una pequeña configuración.

El objetivo de esta actividad es explorar cómo funciona Apache y cómo con sencillas configuraciones podemos servir tanto contenido estático como dinámico.

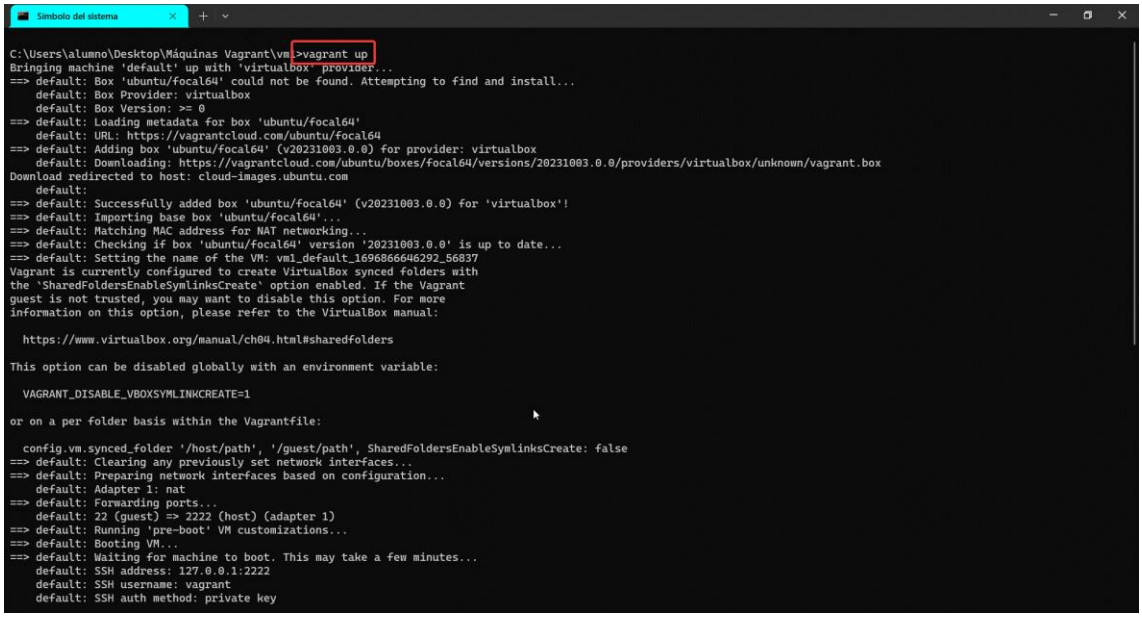
2. ACTIVIDAD PROPUESTA.

2.1 Crea una máquina Ubuntu focal64 con vagrant.

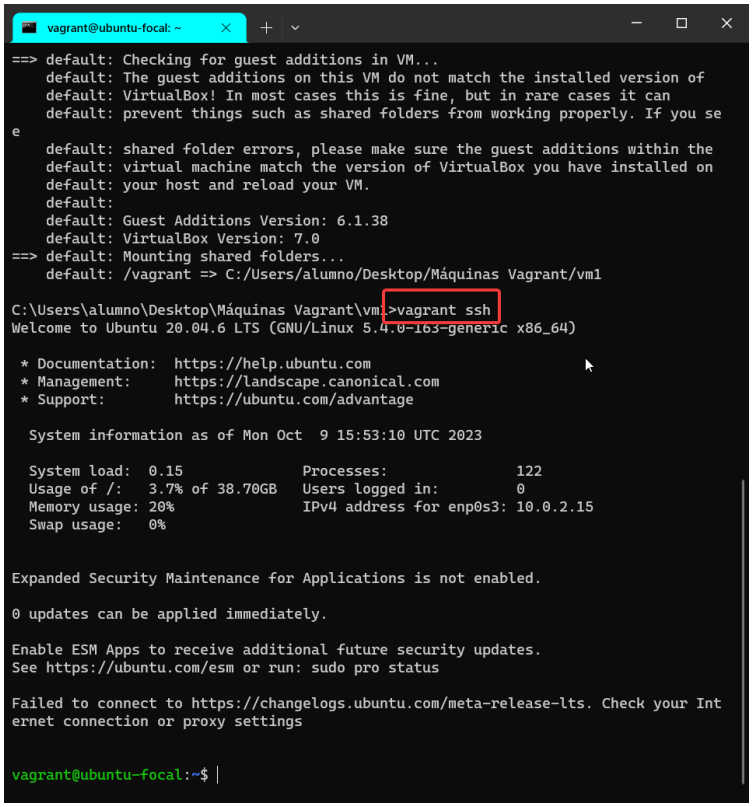
Creamos carpeta box de Ubuntu focal 64 con **vagrant init Ubuntu/focal64** y comprobamos que se ha creado con **dir**:



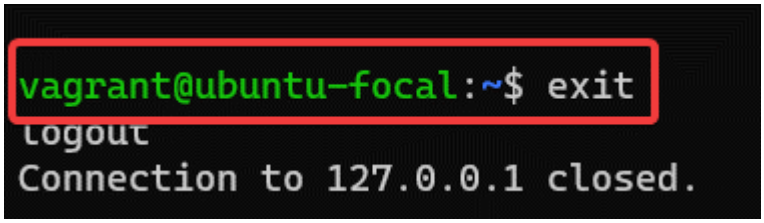
Para descargar la máquina virtual: **vagrant up**.



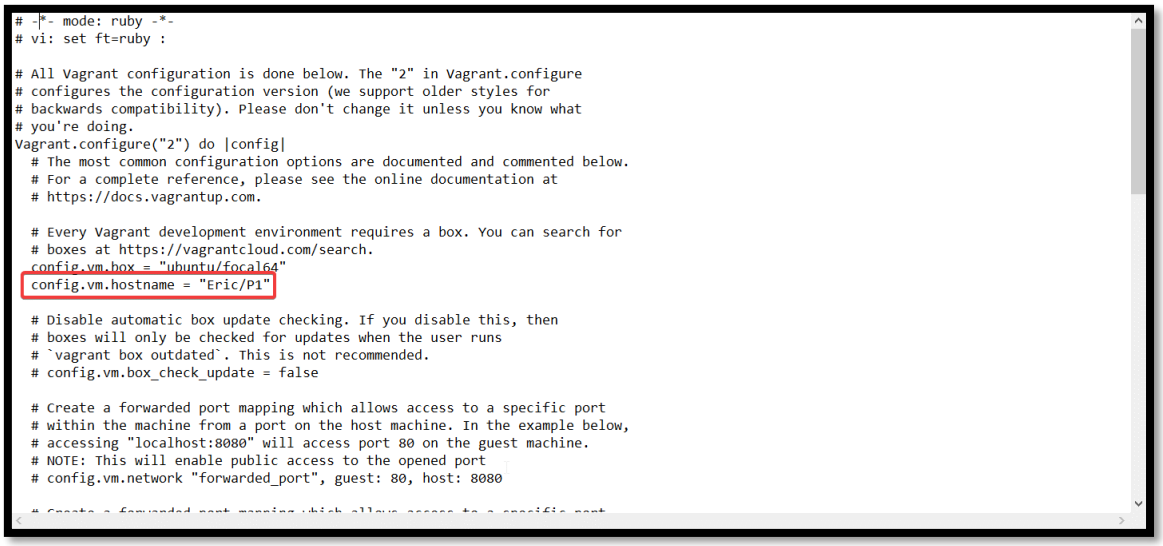
Para iniciar la máquina virtual: **vagrant ssh**



Para salirnos de la máquina: **exit**



Cambiar hostname.

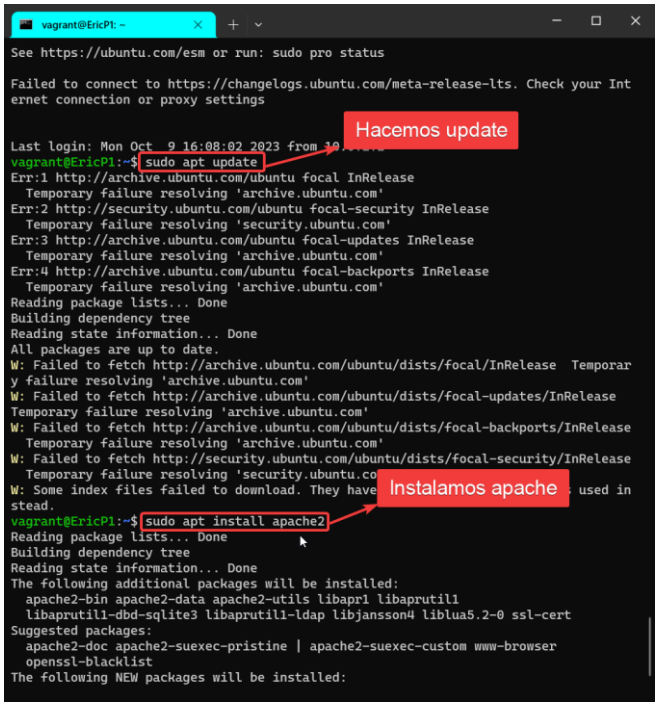


Hacemos un reload para que se apliquen los cambios: **vagrant reload**

2.2 Instalar Apache2 en la máquina virtual y configurar para que el puerto 8080 del host redirija al puerto 80 al guest (Máquina Vagrant)

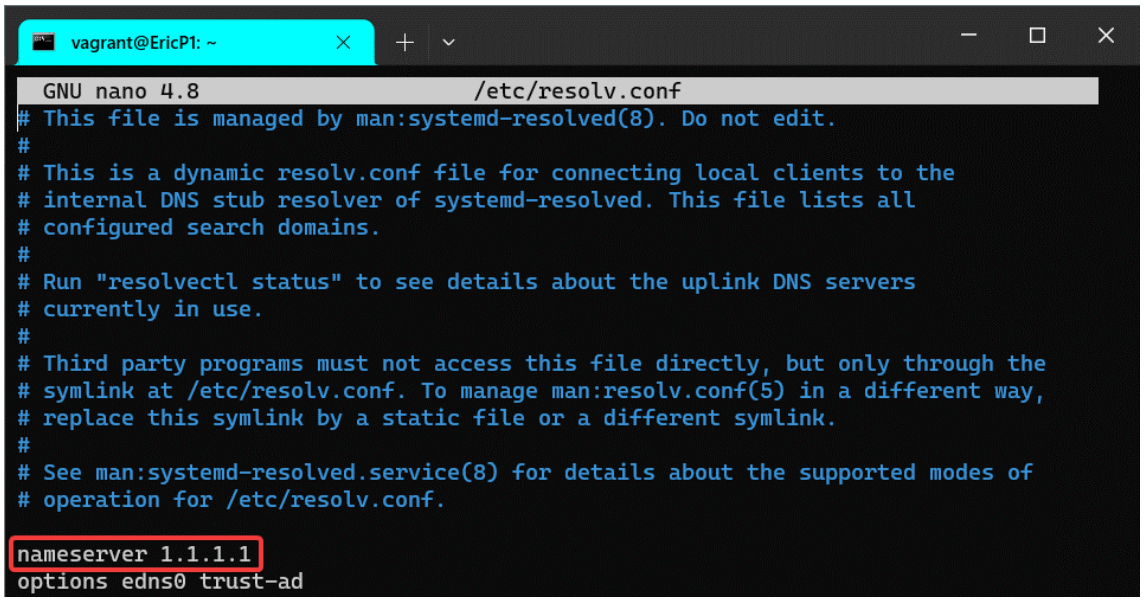
sudo apt update

sudo apt install apache2



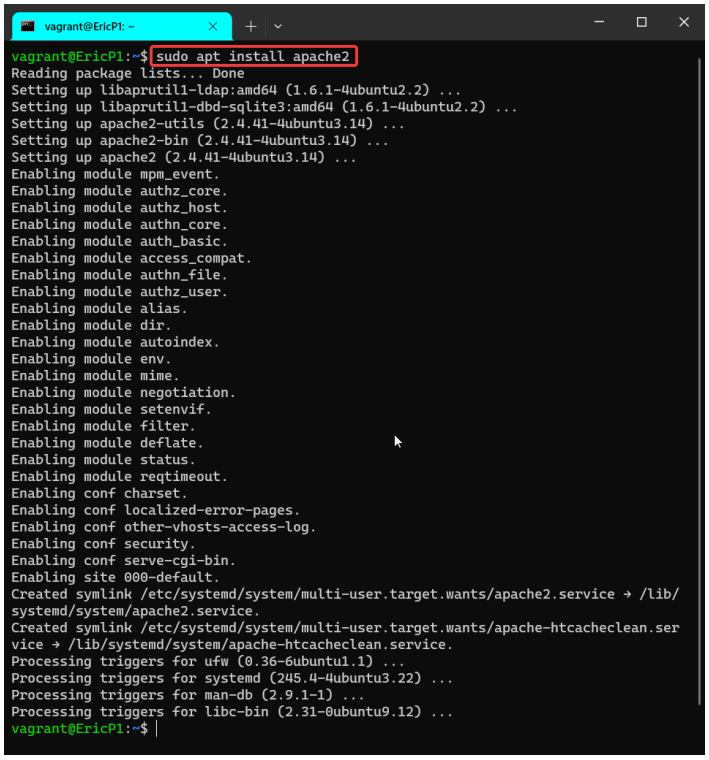
Arreglando problema de DNS.

sudo nano /etc/resolf.conf



Empezamos instalación de Apache2.

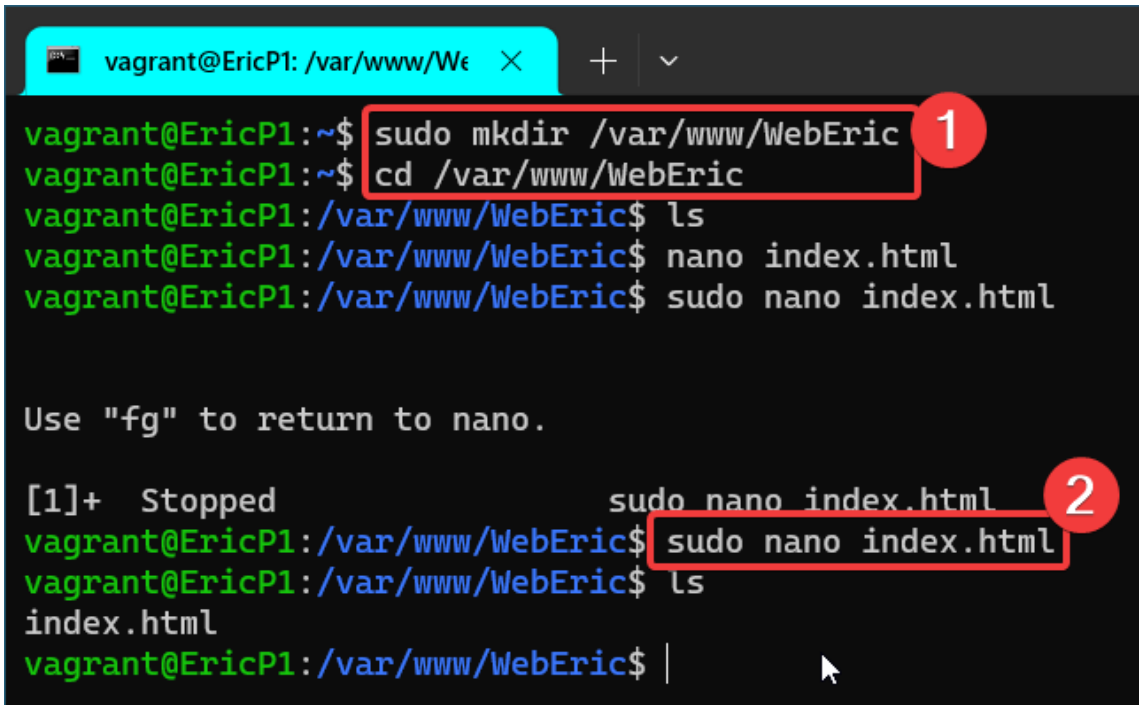
sudo apt install apache2



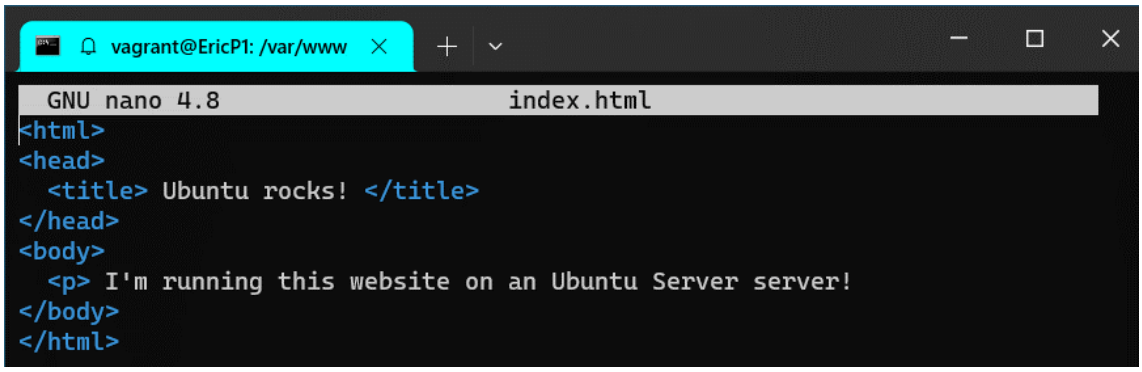
2.3 Configurar un Site en Apache2 con una página de bienvenida estática utilizando una configuración virtualhost propia.

sudo mkdir /var/www/WebEric

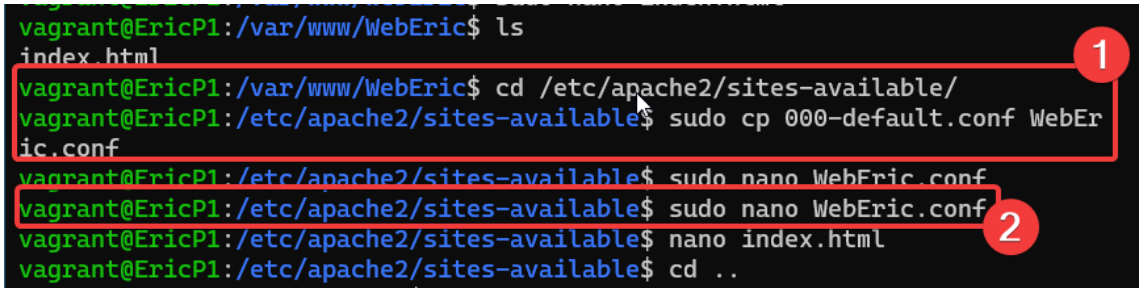
cd /var/www/WebEric



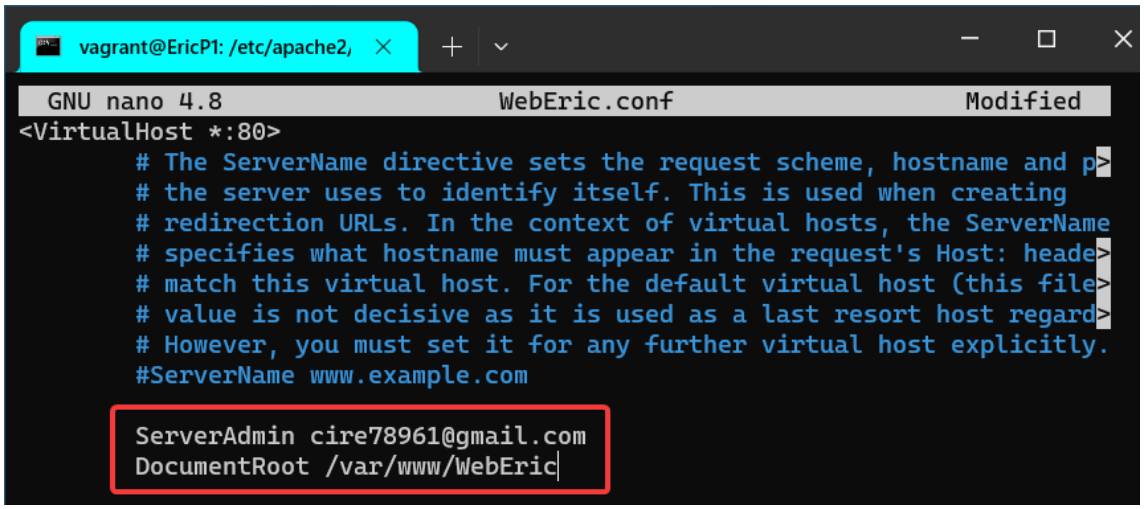
Pegamos un HTML fácil en index.html.



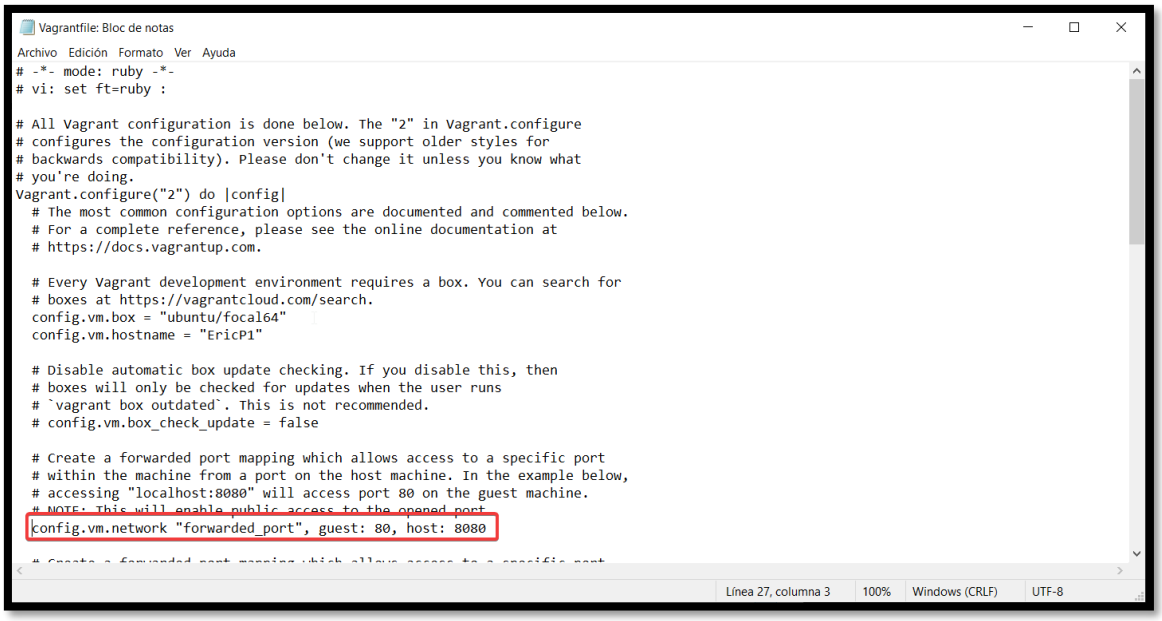
Ahora nos disponemos a copiar todo lo que hay dentro de 000-default.conf en mi archivo de configuración **WebEric.conf**.



En el archivo vamos a cambiar los campos **ServerAdmin** y **DocumentRoot**.



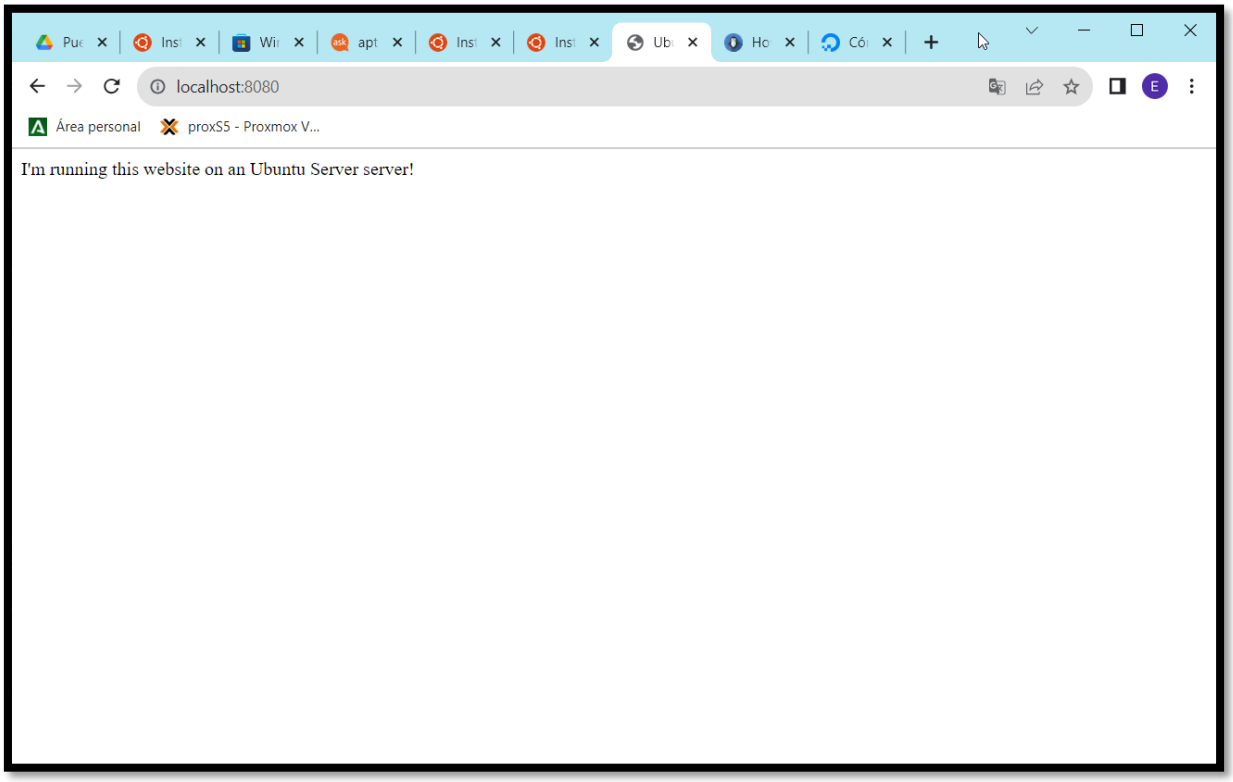
Para poder ver la página en localhost, quitaremos la almohadilla de la línea de texto que vemos a continuación.



Por último, haremos un **sudo systemctl reload apache2**.

```
vagrant@EricP1:/var/www/html$ sudo systemctl reload apache2
vagrant@EricP1:/var/www/html$
```

Y ya podemos ver nuestra página poniendo **localhost:8080**.



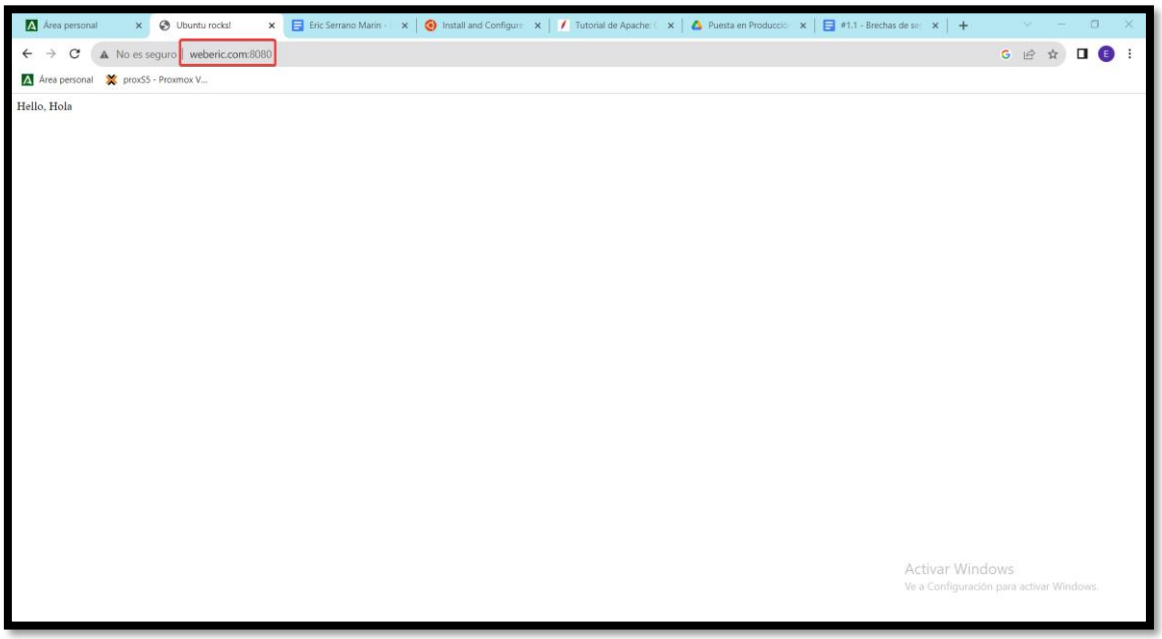
Para poder acceder a nuestra página usando nuestro nombre de dominio, tendremos que cambiar en nuestro Windows el archivo **hosts**.

hosts

C: > Windows > System32 > drivers > etc > hosts

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#   127.0.0.1       localhost
#   ::1             localhost
127.0.0.1 WebEric.com
```

Como podemos observar, ya hemos podido acceder con nuestro nombre de dominio.



2.4 Desarrollar un Script en C que haga un Hola Mundo con un parámetro de entrada que se envié por parámetros en la petición HTTP.

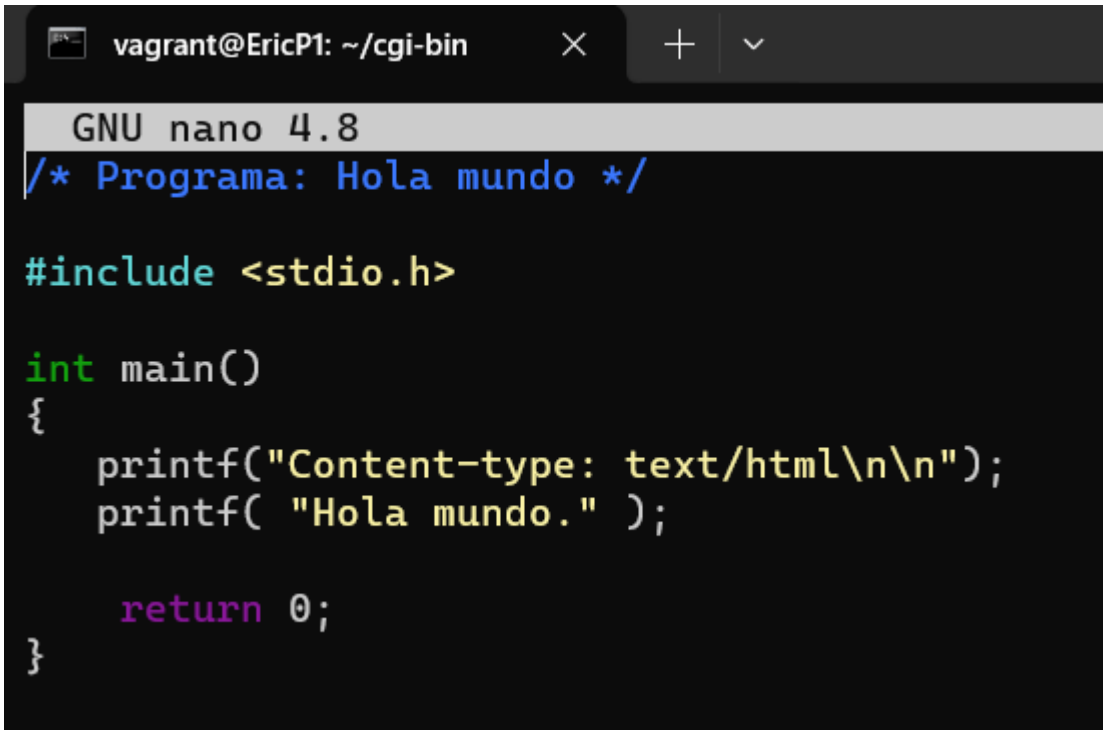
Para este paso he tenido que hacer muchas cosas, de las cuales las voy a enumerar.

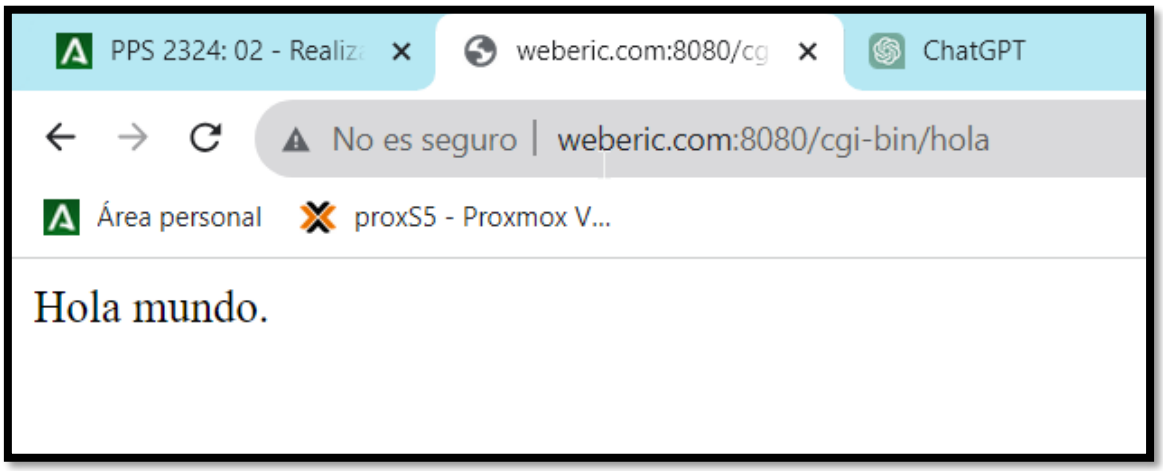
- **sudo a2enmod cgi** se utiliza para habilitar el módulo cgi en el servidor web Apache.
- Configuraciones de archivos: **WebEric.conf**

```
ServerAdmin esermar492@iesmartinezm.com
DocumentRoot /var/www/WebEric
ServerName weberic.com
```

```
ScriptAlias /cgi-bin/ /home/vagrant/cgi-bin/
#LoadModule cgid_module modules/mod_cgid.so
<Directory "/home/vagrant/cgi-bin/">
    Options +ExecCGI
    #AddHandler cgi-script .cgi
    SetHandler cgi-script
    Require all granted
</Directory>
```

- Obviamente también tendremos que hacer sudo chmod +x al script.


vagrant@EricP1: ~/cgi-bin
GNU nano 4.8
/* Programa: Hola mundo */
#include <stdio.h>
int main()
{
printf("Content-type: text/html\n\n");
printf("Hola mundo.");
return 0;
}


PPS 2324: 02 - Realiz
weberic.com:8080/cg
ChatGPT
No es seguro | weberic.com:8080/cgi-bin/hola
Área personal
proxS5 - Proxmox V...
Hola mundo.

2.5 Desarrolla un script en PERL con estructuras selectivas (IF-ELSE). Ejemplo: Mensaje diferente en función de la hora.

sudo apt-get update

sudo apt-get install perl

```
vagrant@EricP1: /
vagrant@EricP1:/$ sudo apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 114 kB in 1s (92.1 kB/s)
Reading package lists... Done
vagrant@EricP1:/$ sudo apt-get install perl
Reading package lists... Done
Building dependency tree
Reading state information... Done
perl is already the newest version (5.30.0-9ubuntu0.4).
perl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
vagrant@EricP1:/$
```

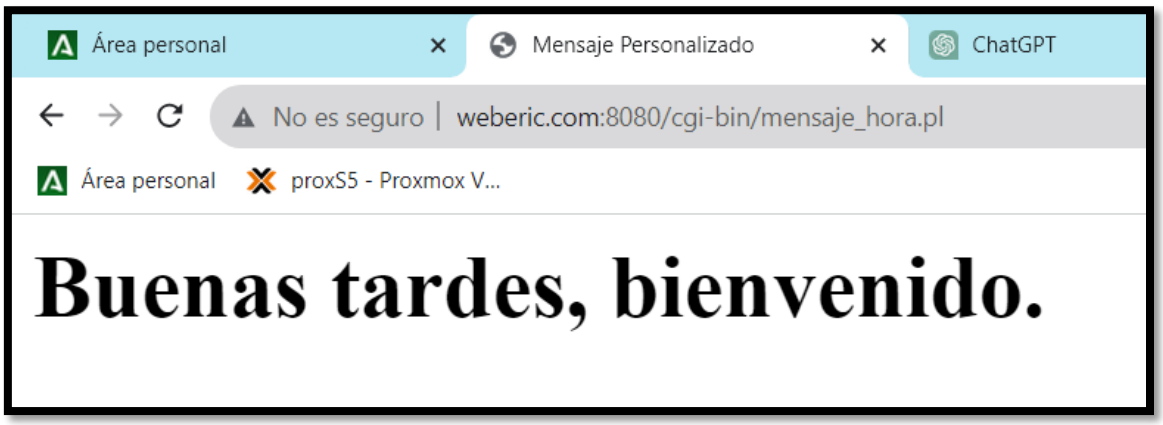
Vamos a darle permisos al archivo .pl: **sudo chmod +x mensaje_hora.pl**

```
vagrant@EricP1:~/cgi-bin$ sudo chmod +x mensaje_hora.pl
```

El código imprime un saludo dependiendo de la hora que sea.

```
vagrant@EricP1: /etc/apache2/
  GNU nano 4.8        /home/vagrant/cgi-bin/mensaje_hora.pl
#!/usr/bin/perl

use strict;
use warnings;

# Obtener la hora actual
my ($sec, $min, $hour) = localtime();

# Aplicar estructura selectiva (IF-ELSE) basada en la hora
if ($hour < 12) {
    $mensaje = "Buenos días, bienvenido.";
} elsif ($hour < 18) {
    $mensaje = "Buenas tardes, bienvenido.";
} else {
    $mensaje = "Buenas noches, bienvenido.";
}

# Imprimir la cabecera HTTP y el contenido HTML
print "Content-Type: text/html\n\n";
print "<html><head><title>Mensaje Personalizado</title></head><body>";
print "<h1>$mensaje</h1>";
print "</body></html>";
```

Como son las 17:00 de la tarde nos dice Buenas tardes.



2.6 Desarrollar un script en Python con estructuras iterativas (WHILE-FOR). Ejemplo imprimir la tabla de multiplicación de la hora del día que sea.

vagrant@EricP1: ~/cgi-bin

GNU nano 4.8 tabla_multiplicar.py

```
#!/usr/bin/env python3

import time

# Obtener la hora actual
hora_actual = time.localtime().tm_hour

# Mensaje predeterminado
mensaje = f"Tabla de multiplicar para la hora {hora_actual}:<br>"

# Estructura WHILE para imprimir la tabla de multiplicar
contador = 1
while contador <= 10:
    resultado = hora_actual * contador
    mensaje += f"{hora_actual} x {contador} = {resultado}<br>"
    contador += 1

# Añadir un salto de línea entre las dos tablas
mensaje += "<br>"

# Estructura FOR para imprimir la tabla de multiplicar de 2 en 2
mensaje += f"Tabla de multiplicar de 2 en 2 para la hora {hora_actual}:<br>"
for contador in range(1, 11, 2):
    resultado = hora_actual * contador
    mensaje += f"{hora_actual} x {contador} = {resultado}<br>"

# Imprimir el mensaje
print("Content-type: text/html\n\n")
print("<html><head><title>Tabla de Multiplicar</title></head><body>")
print(f"<h2>{mensaje}</h2>")
print("</body></html>")
```

Tenemos que darle permisos de ejecución con sudo chmod +x tabla_multiplicar.py.