

# *Práctica 4: Servidor web Apache*

Manuel Peinado Cuenca

[i22pecum@uco.es](mailto:i22pecum@uco.es)

<b>Ejercicio 1</b>	<b>4</b>
Enunciado	4
Resolución	4
<b>Ejercicio 2</b>	<b>6</b>
Enunciado	6
Resolución	6
<b>Ejercicio 3</b>	<b>7</b>
Enunciado	7
Resolución	7
<b>Ejercicio 4</b>	<b>9</b>
Enunciado	9
Resolución	9
<b>Ejercicio 5</b>	<b>11</b>
Enunciado	11
Resolución	11
<b>Ejercicio 6</b>	<b>14</b>
Enunciado	14
Resolución	14
<b>Ejercicio 7</b>	<b>19</b>
Enunciado	19
Resolución	19
<b>Ejercicio 8</b>	<b>21</b>
Enunciado	21
Resolución	21
<b>Ejercicio 9</b>	<b>22</b>
Enunciado	22
Resolución	22
<b>Ejercicio 10</b>	<b>24</b>
Enunciado	24
Resolución	24
<b>Ejercicio 11</b>	<b>28</b>
Enunciado	28
Resolución	28
<b>Ejercicio 12</b>	<b>30</b>
Enunciado	30
Resolución	30
<b>Ejercicio 13</b>	<b>31</b>
Enunciado	31
Resolución	31
<b>Ejercicio 14</b>	<b>34</b>
Enunciado	34

Resolución.....	34
<b>Ejercicio 15.....</b>	<b>37</b>
Enunciado.....	37
Resolución.....	37
<b>Ejercicio 16.....</b>	<b>41</b>
Enunciado.....	41
Resolución.....	41

# Ejercicio 1

## Enunciado

Explica qué hace cada una de las líneas del bash script de instalación de Apache. Puedes colocar los comentarios dentro del propio archivo.

## Resolución

La líneas del script son las siguientes:

**mkdir \$HOME/tmp1** → Crea en el directorio home del usuario que ejecuta el script un directorio que se llame tmp1

**cd \$HOME/tmp1** → Abre el directorio que se acaba de crear

**wget -O httpd-2.4.59.tar.bz2 <https://dlcdn.apache.org/httpd/httpd-2.4.59.tar.bz2>** → Descarga el archivo comprimido de Apache HTTP Server versión 2.4.59 y lo guarda con el nombre httpd-2.4.59.tar.bz2

**tar jxvf httpd-2.4.59.tar.bz2** → Descomprime el archivo httpd-2.4.59.tar.bz2

**wget -O apr-1.6.5.tar.bz2 <https://dlcdn.apache.org/apr/apr-1.6.5.tar.bz2>** → Descarga el archivo comprimido de APR versión 1.6.5.

**wget -O apr-util-1.6.3.tar.bz2 <https://dlcdn.apache.org/apr/apr-util-1.6.3.tar.bz2>** → Descarga el archivo comprimido de APR-util versión 1.6.3.

**tar jxvf apr-1.6.5.tar.bz2** → Descomprime el archivo apr-1.6.5.tar.bz2

**tar jxvf apr-util-1.6.3.tar.bz2** → Descomprime el archivo apr-util-1.6.3.tar.bz2

**sudo apt-get update** → Actualiza la lista de paquetes disponibles para descargar

**sudo apt-get install build-essential** → Instala el paquete build-essential, que incluye herramientas necesarias para compilar software

**mv apr-1.6.5 ./httpd-2.4.59/src/lib/apr** → Mueve la carpeta apr-1.6.5 al directorio httpd-2.4.59/src/lib/apr

**mv apr-util-1.6.3 ./httpd-2.4.59/src/lib/apr-util** → Mueve la carpeta apr-util-1.6.3 al directorio httpd-2.4.59/src/lib/apr-util

**sudo apt-get install libpcre3 libpcre3-dev** → Instala las bibliotecas de expresiones regulares compatibles con Perl y sus archivos de desarrollo

**sudo apt-get install libexpat1-dev** → Instala la biblioteca expat y sus archivos de desarrollo

**cd httpd-2.4.59** → Entra el directorio httpd-2.4.59

**./configure --prefix=\$HOME/httpd --with-included-apr** → Configura la compilación de Apache HTTP Server con la ruta de instalación en el directorio ~/httpd y utilizando las versiones de APR y APR-util incluidas.

**sudo make -j4** → Compila Apache HTTP Server utilizando 4 núcleos de CPU simultáneamente

**sudo make install** → Instala Apache HTTP Server

**sudo rm -rf \$HOME/tmp1** → Elimina recursivamente el directorio temporal tmp1 y todo su contenido

## Ejercicio 2

### Enunciado

Recorre las carpetas del servidor observando qué se guarda en ellas. Necesitarás acceder al archivo `httpd.conf` para modificar la configuración de Apache.

### Resolución

Las carpetas del servidor son las siguientes:

**bin** → Esta carpeta contiene los ejecutables del servidor web Apache

**cgi-bin** → Esta carpeta es utilizada para almacenar scripts CGI (Common Gateway Interface)

**error** → Aquí se encuentran los archivos de configuración relacionados con la gestión de errores

**icons** → En esta carpeta se almacenan los iconos predeterminados que se utilizan para representar diversos tipos de archivos y directorios en las páginas de listado de directorios

**lib** → Esta carpeta contiene las bibliotecas compartidas necesarias para que el servidor web Apache funcione correctamente

**man** → Aquí se encuentran las páginas del manual para el servidor Apache

**modules** → Esta carpeta contiene los módulos que pueden ser cargados dinámicamente por el servidor Apache para extender su funcionalidad

**build** → Esta carpeta contiene archivos relacionados con la construcción y compilación del servidor Apache, como scripts de configuración y archivos de registro

**conf** → Aquí se encuentran los archivos de configuración principales del servidor Apache

**htdocs** → Esta carpeta es el directorio raíz del servidor web, donde se almacenan los archivos que serán servidos por Apache

**include** → Aquí se pueden encontrar archivos de inclusión que son utilizados por el servidor Apache para configuraciones específicas

**logs** → Esta carpeta contiene los archivos de registro generados por el servidor Apache

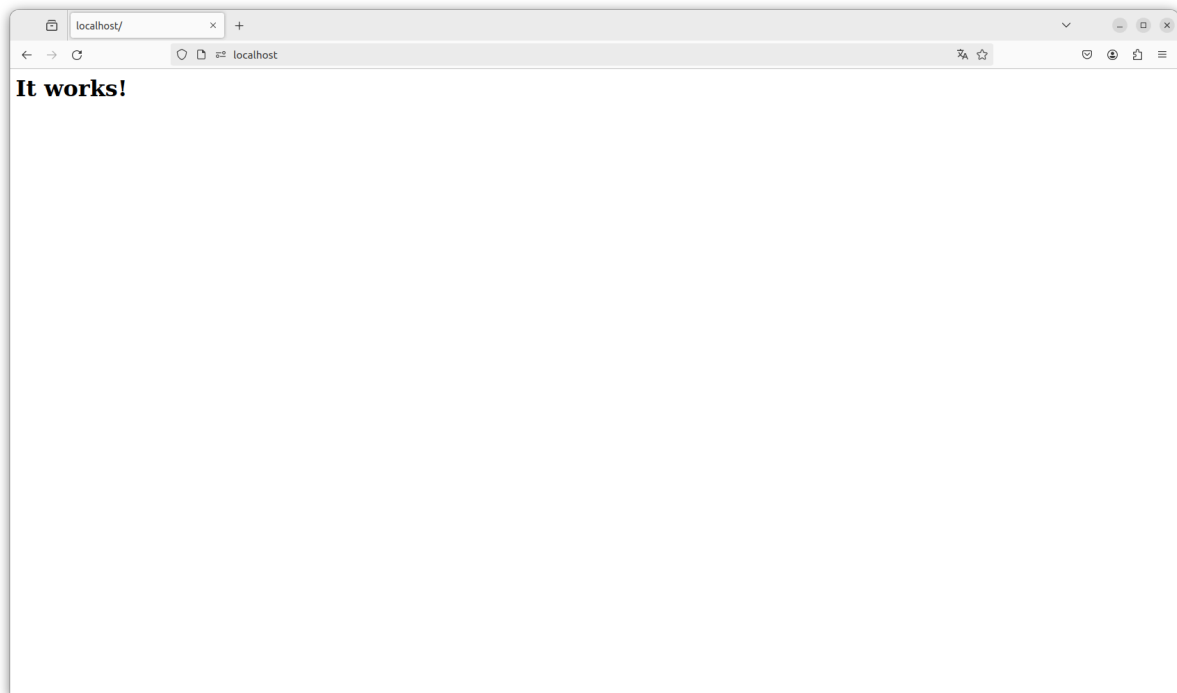
**manual** → Aquí se encuentra la documentación completa del servidor Apache

## Ejercicio 3

### Enunciado

Si has instalado el servidor Apache en Ubuntu con el script de Moodle, inicia y para el servidor Apache utilizando el ejecutable `apachectl` (carpeta `/bin`) con las opciones `start` y `stop`. Recuerda que tendrás que ejecutar `apachectl restart` cada vez que hagas un cambio en la configuración. Si has instalado Apache utilizando el paquete XAMPP basta que utilices la aplicación gráfica para encender, apagar o reiniciar el servidor Apache. Los leds de estado pasarán a verde. Abre el navegador con `http://localhost` y captura la pantalla. Si quiero que el servidor web Apache escuche en otro puerto, por ejemplo, el 8080, ¿qué opción del archivo configuración debo modificar? Haga captura de pantalla de `http://localhost:8080`.

### Resolución

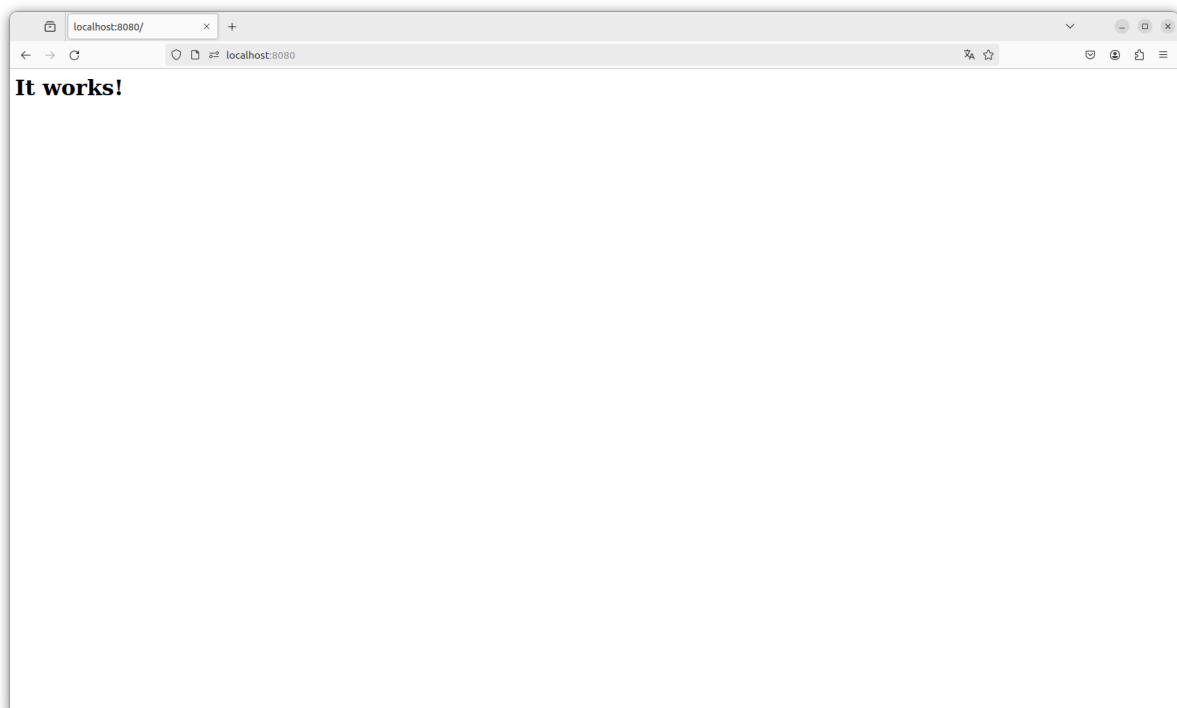


Para cambiar el puerto hay que abrir el fichero `httpd.conf` y modificar la directiva “Listen” en este caso ponemos “Listen 8080” para que escuche el puerto 8080.

```
root@Ubuntu: ~/httpd/conf
GNU nano 6.2 httpd.conf
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 8080

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace  ^U Paste     ^J Justify  ^_ Go To Line
```





# Ejercicio 4

## Enunciado

La carpeta raíz del servidor Apache es /htdocs. Esta es la carpeta donde primero busca Apache el archivo index.html. Nos bajamos un archivo index.html de Internet y lo colocamos en la raíz. Comprobar que ahora se muestra este archivo en lugar del index.html. Puedes buscar en Internet archivos básicos de HTML (Mirad lo que se puede hacer con HTML5: <https://www.tutorialrepublic.com/html-examples.php>). Nota: si estamos con XAMPP y aún no hemos creado el index.html se ejecutará el index.php que ya existe en el directorio raíz y que redirige a la carpeta /dashboard.

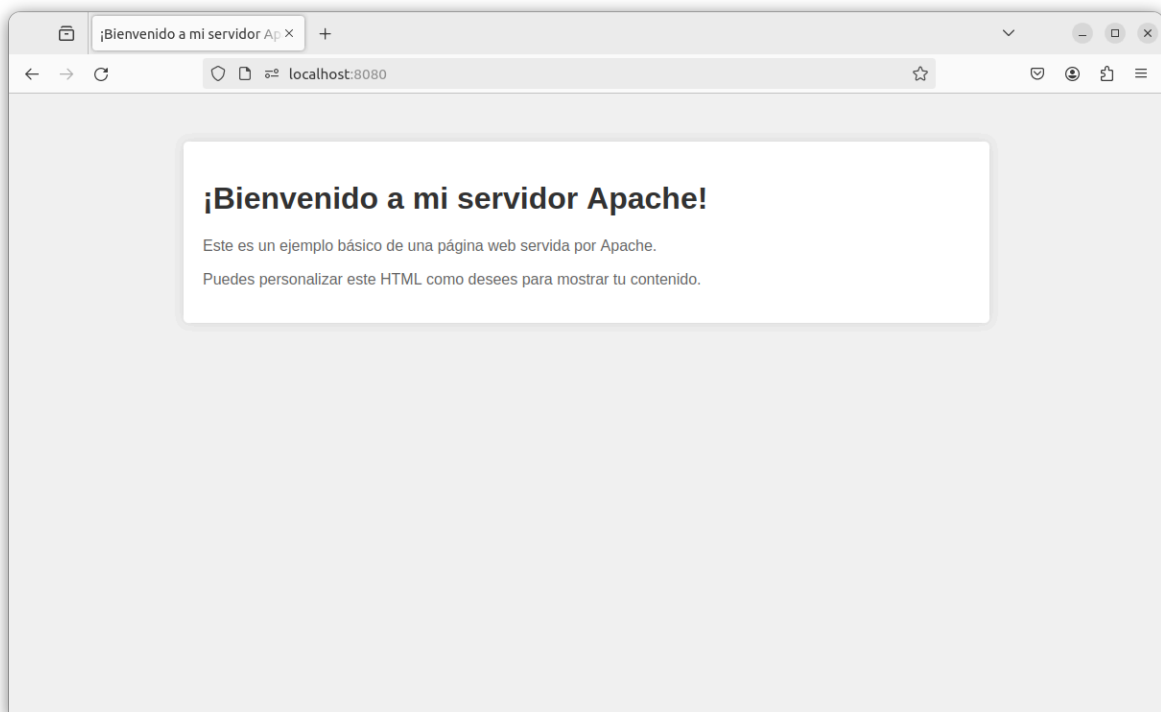
## Resolución

A continuación le he pedido un código simple HTML a Chat GPT para cambiar el index.html, me ha dado el código siguiente:

```
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>¡Bienvenido a mi servidor Apache!</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f0f0f0;
    }
    .container {
      max-width: 800px;
      margin: 50px auto;
      padding: 20px;
      background-color: #fff;
      border-radius: 5px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
    h1 {
      color: #333;
    }
    p {
      color: #666;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>¡Bienvenido a mi servidor Apache!</h1>
    <p>Este es un ejemplo de un archivo index.html creado con Chat GPT.</p>
  </div>
</body>
</html>
```

```
</style>
</head>
<body>
  <div class="container">
    <h1>¡Bienvenido a mi servidor Apache!</h1>
    <p>Este es un ejemplo básico de una página web servida por Apache.</p>
    <p>Puedes personalizar este HTML como desees para mostrar tu contenido.</p>
  </div>
</body>
</html>
```

Que hace que mi servidor de apache se vea así:



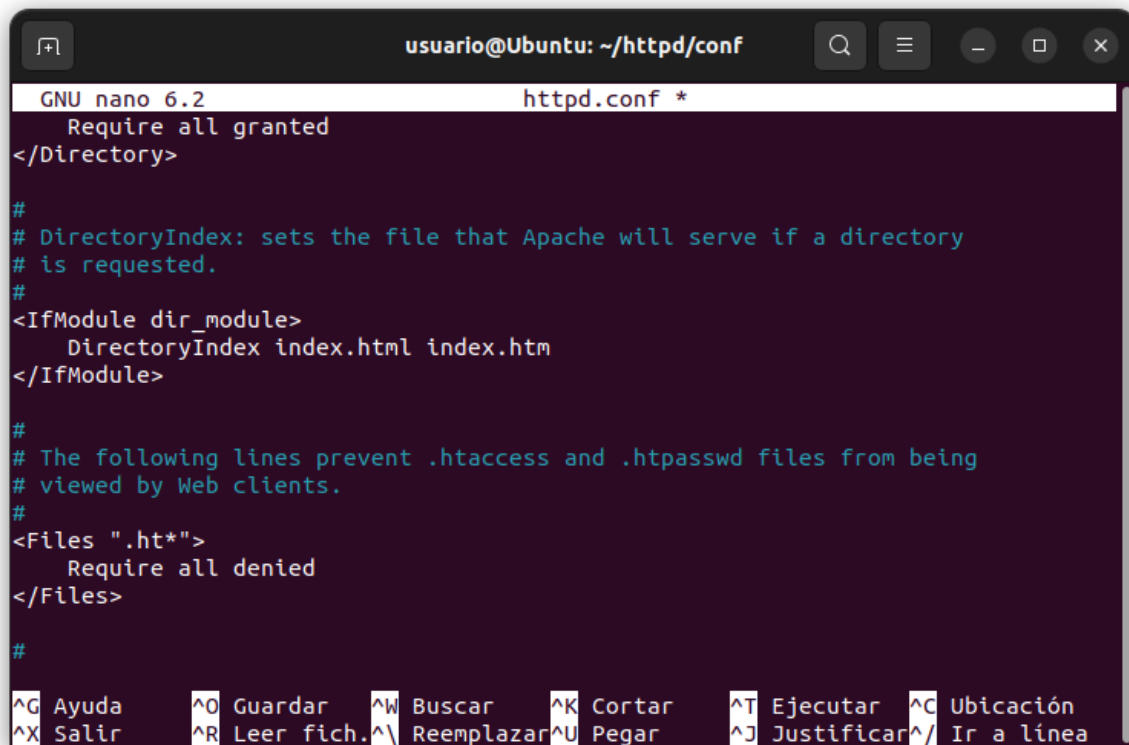
# Ejercicio 5

## Enunciado

Cambia el nombre de los archivos índice (archivos que servirá Apache cuando el cliente sólo especifique un directorio). Normalmente es index.html, añadir como posibilidad index.htm y probar si funciona. ¿Qué prioridad se utiliza si tenemos más de un archivo índice?

## Resolución

Para modificar el nombre de los archivos índice nos vamos al fichero httpd.conf, dentro de este fichero nos vamos a la directiva “DirectoryIndex” y ahí añadimos los nombres que queremos que acepte



```
GNU nano 6.2 httpd.conf *
    Require all granted
</Directory>

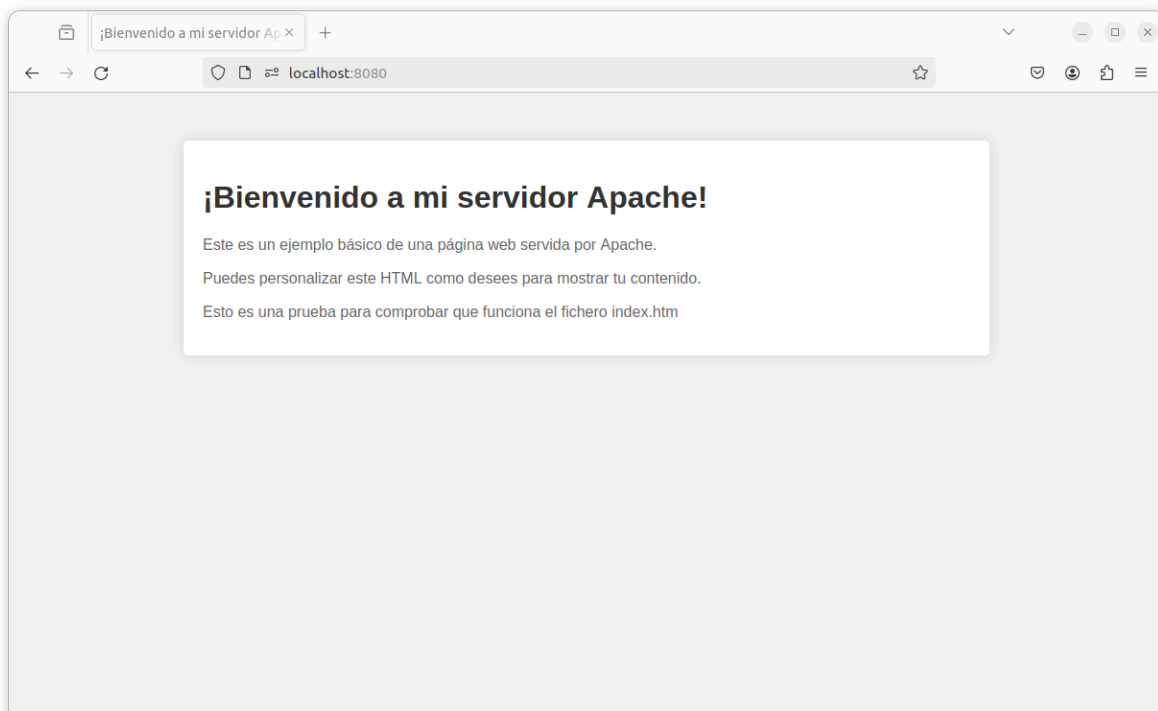
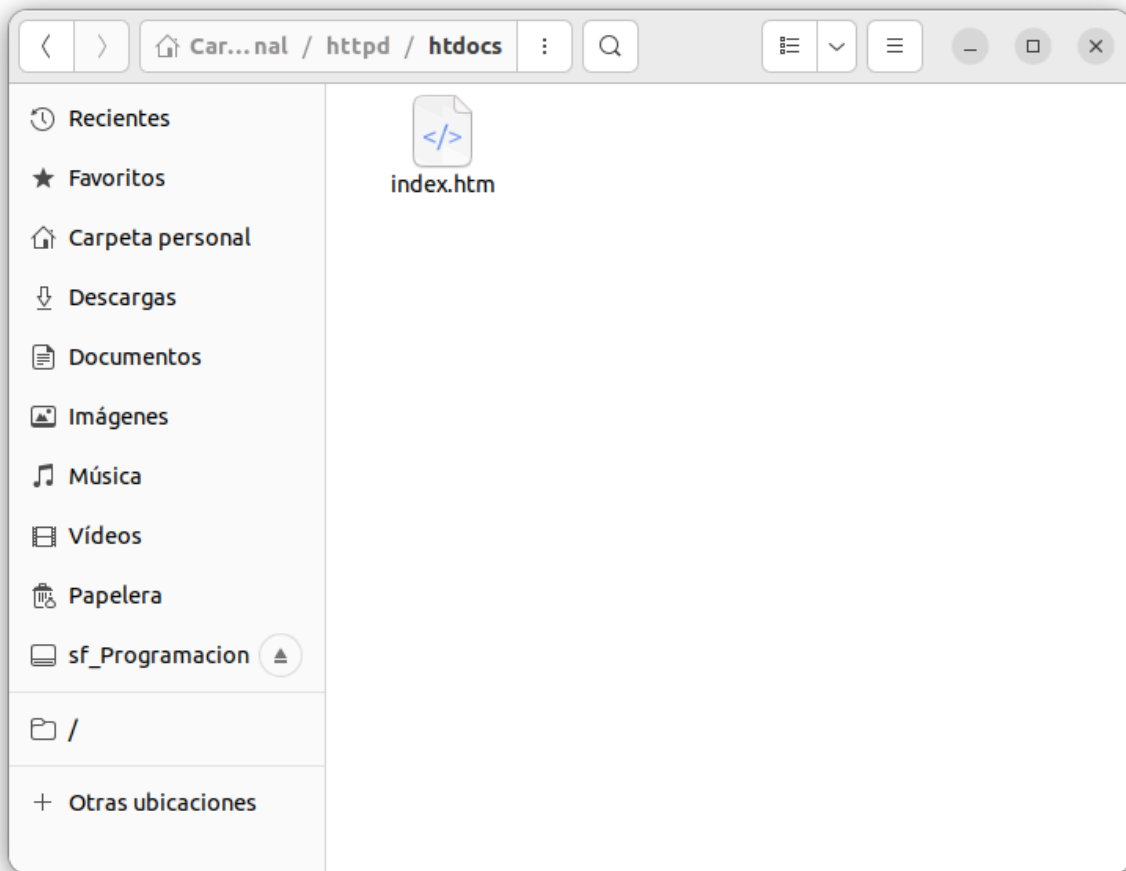
#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
    DirectoryIndex index.html index.htm
</IfModule>

#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ".ht*">
    Require all denied
</Files>

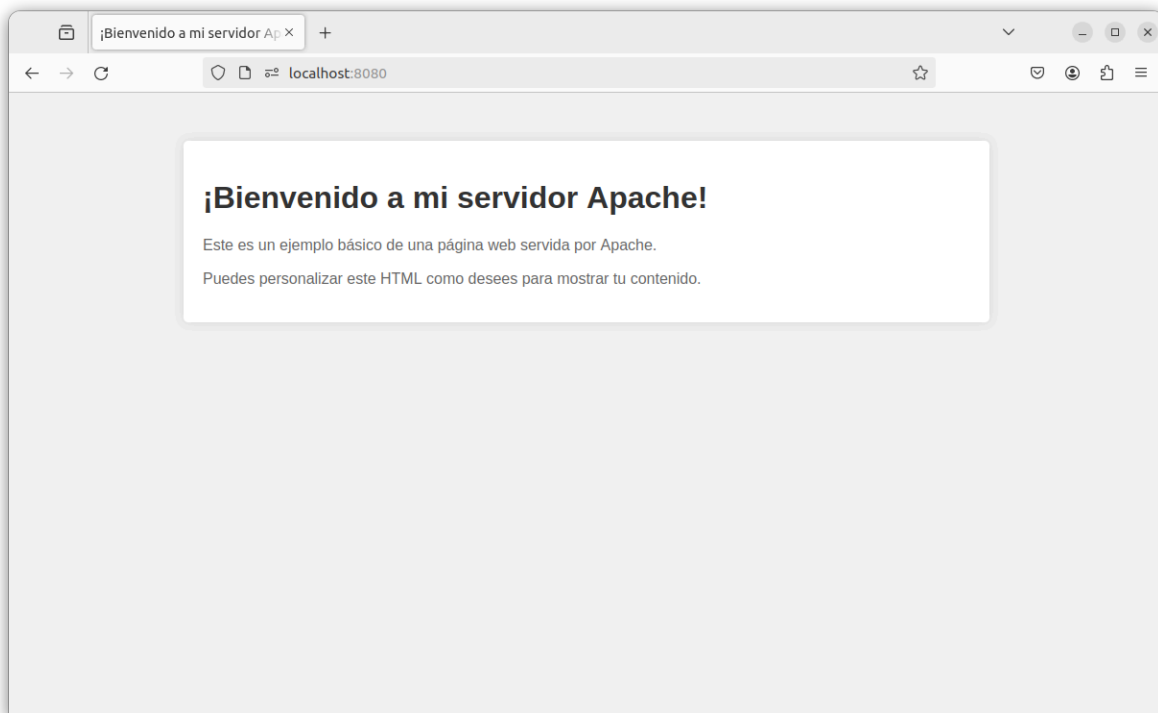
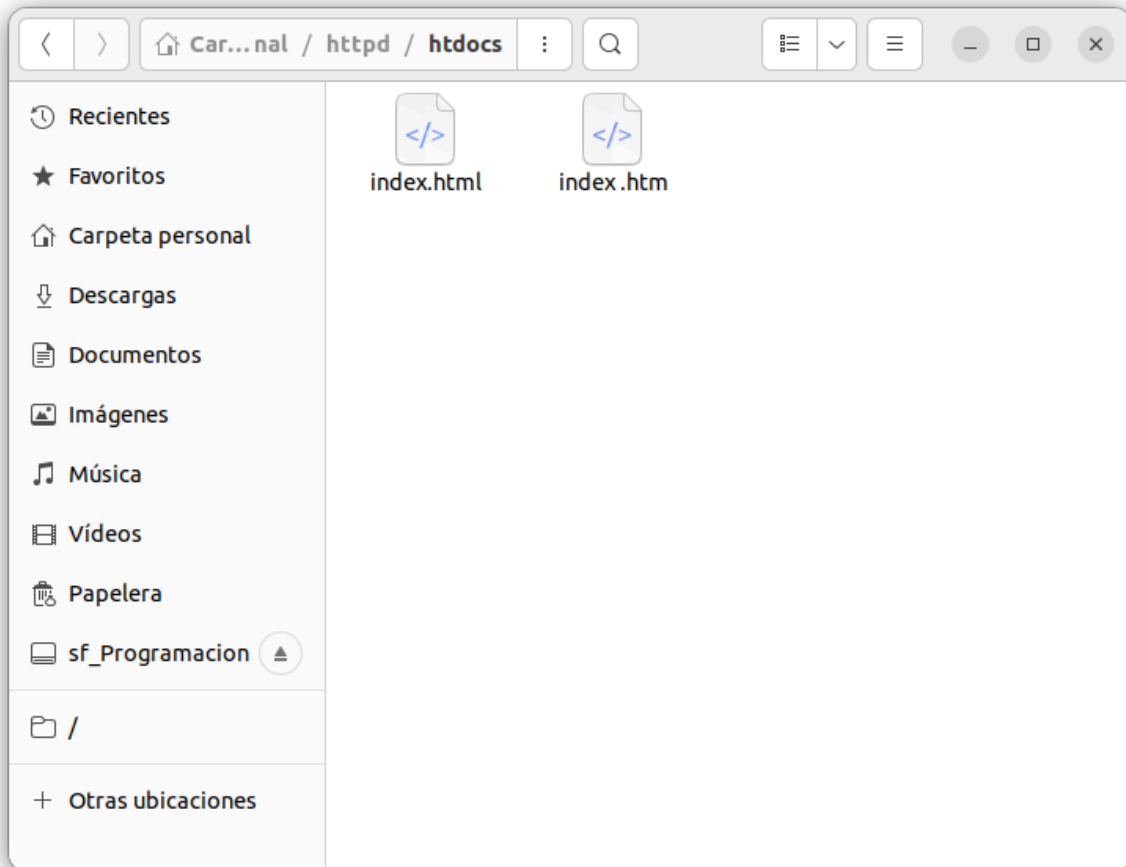
#

^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^\ Reemplazar  ^U Pegar      ^J Justificar ^_ Ir a línea
```

Comprobamos que funciona cambiando el nombre al fichero index.html por index.htm



Comprobamos que funciona correctamente, ahora volvemos a crear el fichero index.html para ver cual de los dos se ejecutaría



Y comprobamos que se nos ejecuta index.html, esto es porque se ejecutan en el orden en el que están puestos en la directiva “DirectoryIndex”.

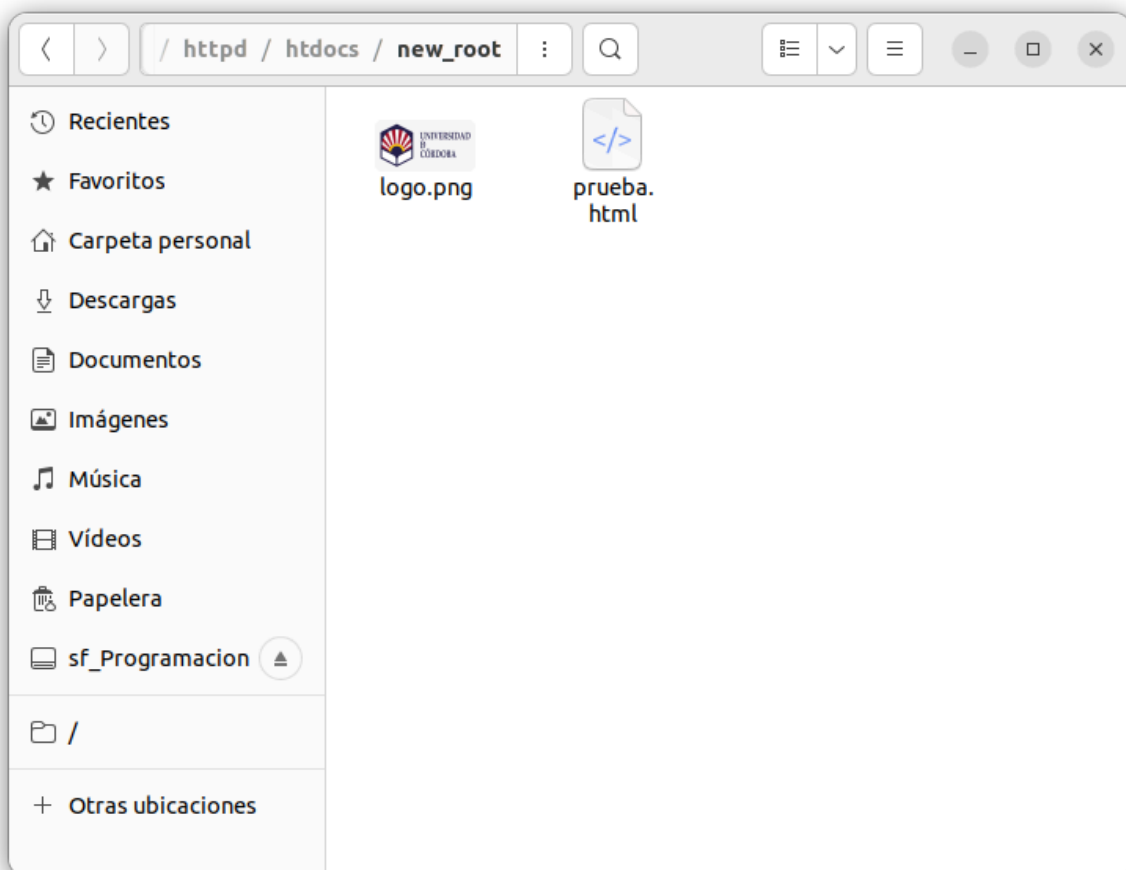
# Ejercicio 6

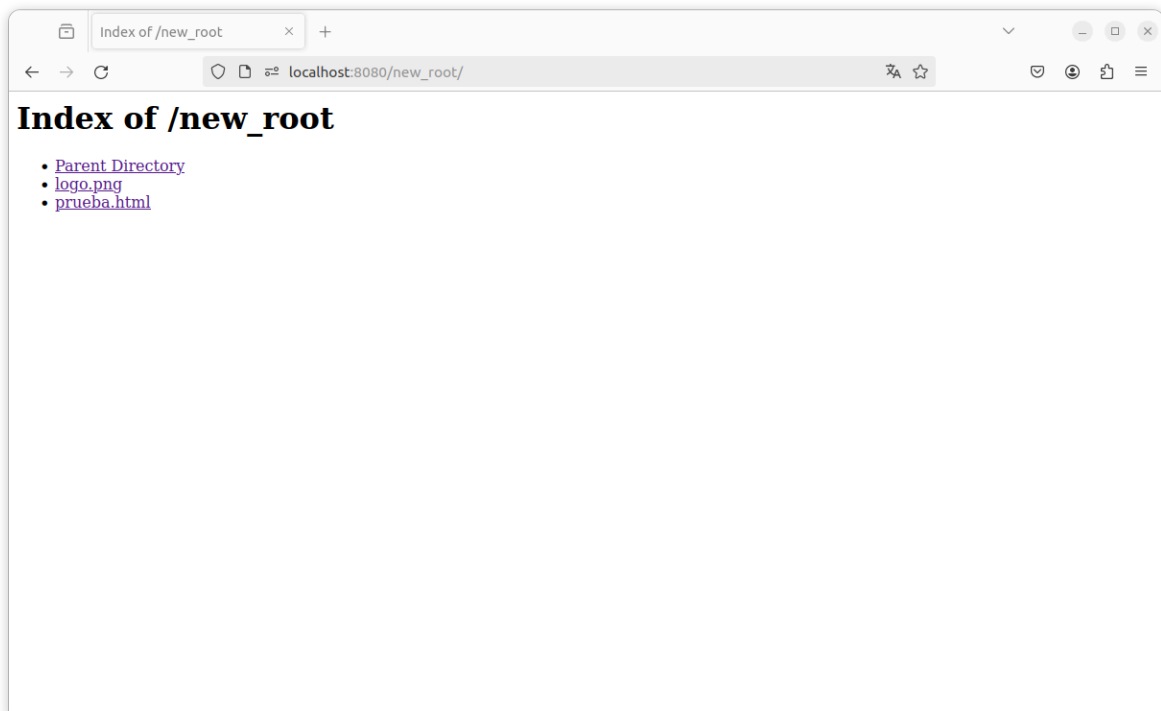
## Enunciado

¿Qué opción de qué directiva es la encargada de permitir mostrar el contenido de un directorio aunque éste no contenga ningún archivo índice? Crea un nuevo directorio llamado `new_root` dentro de la carpeta `/htdocs`, incluye un archivo cualquiera (que no sea `index.html`) y una imagen y prueba a utilizar esta opción para prohibir o no el listado de archivos en dicha carpeta. Para probarlo tendrás que acceder a `localhost/new_root`. Agrega la Descripción “Cuadro de mando” al archivo de imagen que está en la carpeta (pistas: `Directory` y `Options Indexes`..)

## Resolución

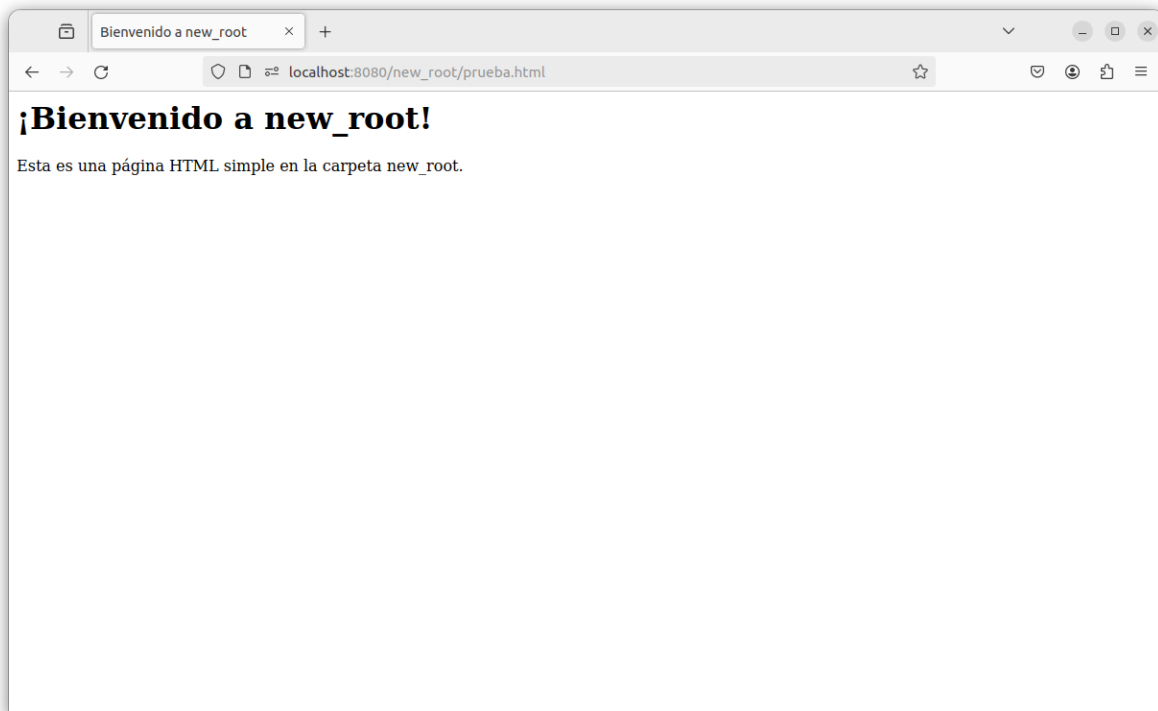
Es la opción “Options Indexes” de la directiva “Directory”  
Creamos el directorio `new_root` e introducimos un fichero `html` y una imagen.





Y si probamos a abrir los enlaces simbólicos que nos aparecen.





Y para prohibir el listado habría que irse al fichero httpd.conf y en la directiva “Directory” modificar la línea “Options” y ponerle un signo menos delante a “Indexes” y otro a “FollowSymLinks”

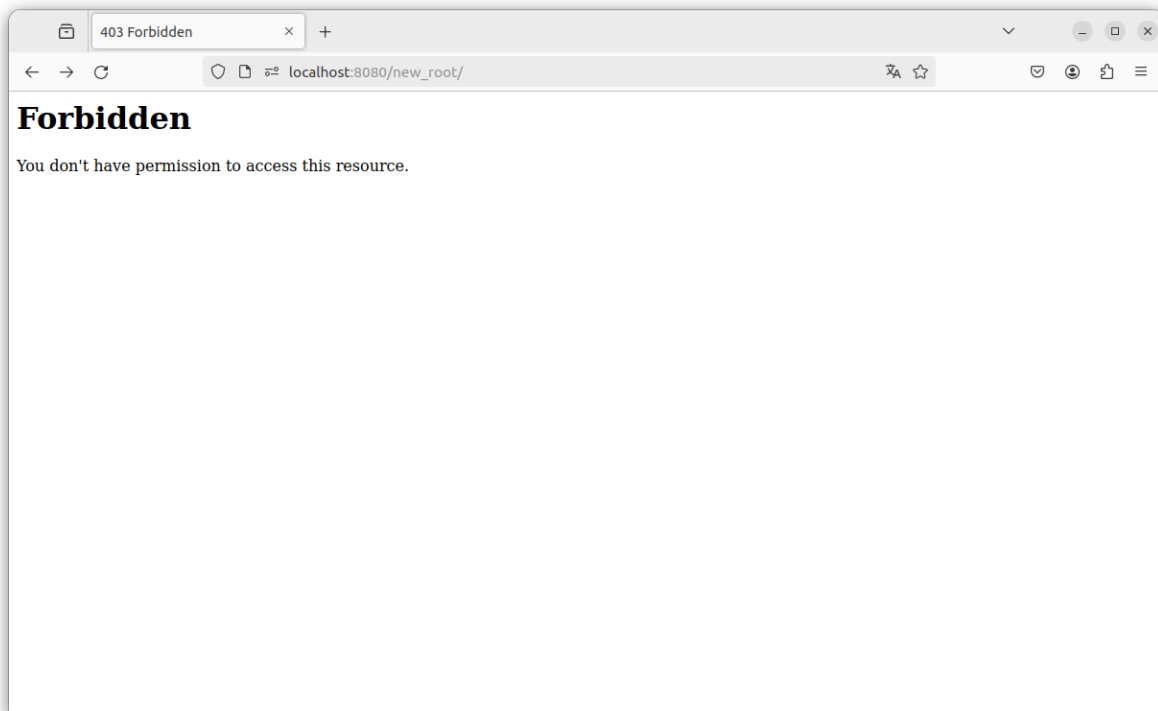
A screenshot of a terminal window with a dark background. The title bar shows "usuario@Ubuntu: ~/httpd/conf". The window title is "GNU nano 6.2 httpd.conf". The content shows the configuration for the `Directory "/home/usuario/httpd/htdocs">` block. The `Options` line is set to `-Indexes -FollowSymLinks`.

```
#
DocumentRoot "/home/usuario/httpd/htdocs"
<Directory "/home/usuario/httpd/htdocs">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important. Please see
    # http://httpd.apache.org/docs/2.4/mod/core.html#options
    # for more information.
    #
    Options -Indexes -FollowSymLinks

    #
    # AllowOverride controls what directives may be placed in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
```

At the bottom, there is a row of keyboard shortcuts: `^G` Ayuda, `^O` Guardar, `^W` Buscar, `^K` Cortar, `^T` Ejecutar, `^C` Ubicación, `^X` Salir, `^R` Leer fich., `^I` Reemplazar, `^U` Pegar, `^J` Justificar, `^/` Ir a línea.

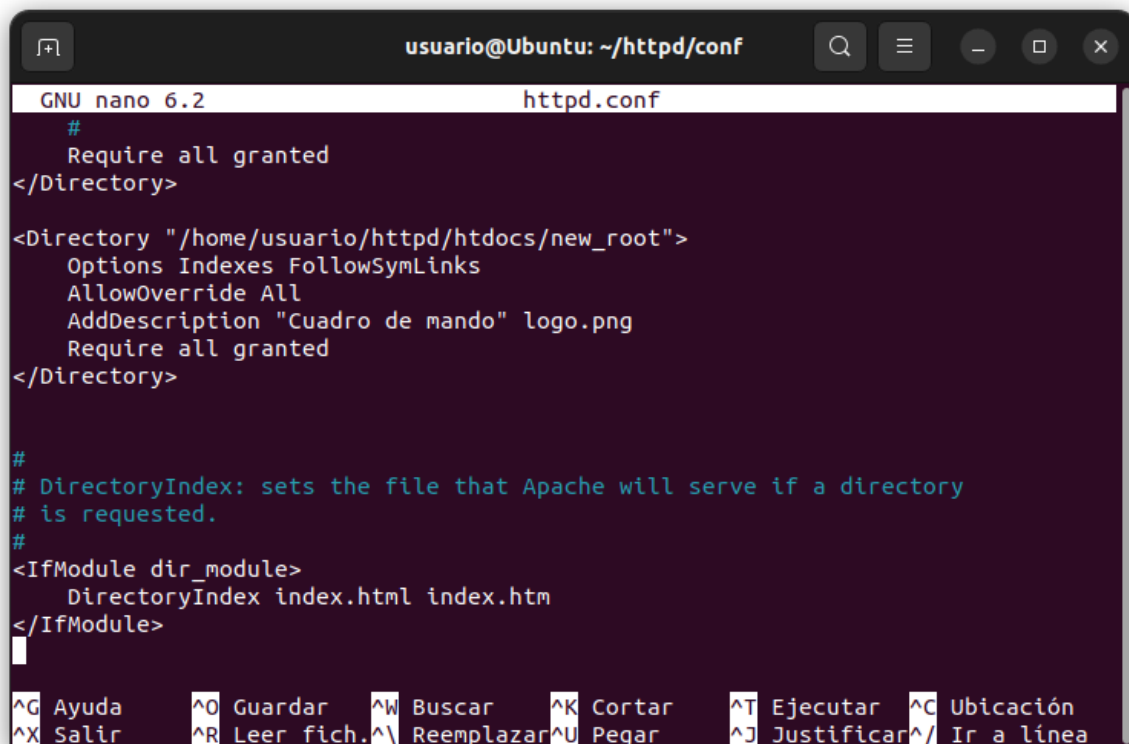




A continuación para añadir la descripción a un fichero, en primer lugar tenemos que descomentar la línea “Include conf/extra/httpd-autoindex.conf” del fichero httpd.conf

A screenshot of a terminal window titled "usuario@Ubuntu: ~/httpd/conf". It shows the GNU nano 6.2 text editor editing the file httpd.conf. The editor's content includes several commented-out lines and one active line: "# Server-pool management (MPM specific)", "#Include conf/extra/httpd-mpm.conf", "# Multi-language error messages", "#Include conf/extra/httpd-multilang-errordoc.conf", "# Fancy directory listings", "Include conf/extra/httpd-autoindex.conf", "# Language settings", "#Include conf/extra/httpd-languages.conf", "# User home directories", "#Include conf/extra/httpd-userdir.conf", "# Real-time info on requests and configuration", and "#Include conf/extra/httpd-info.conf". A status bar at the bottom indicates "537 líneas escritas". A keyboard shortcuts menu is visible at the bottom of the terminal, listing various commands like Ayuda, Guardar, Buscar, Cortar, Ejecutar, Ubicación, Salir, Leer fich., Reemplazar, Pegar, Justificar, and Ir a línea.

A continuación añadimos al fichero httpd.conf en la directiva del directorio que queremos modificar añadimos la opción “AddDescription” con la descripción que queremos añadir y el fichero al que queremos que le aparezca dicha descripción

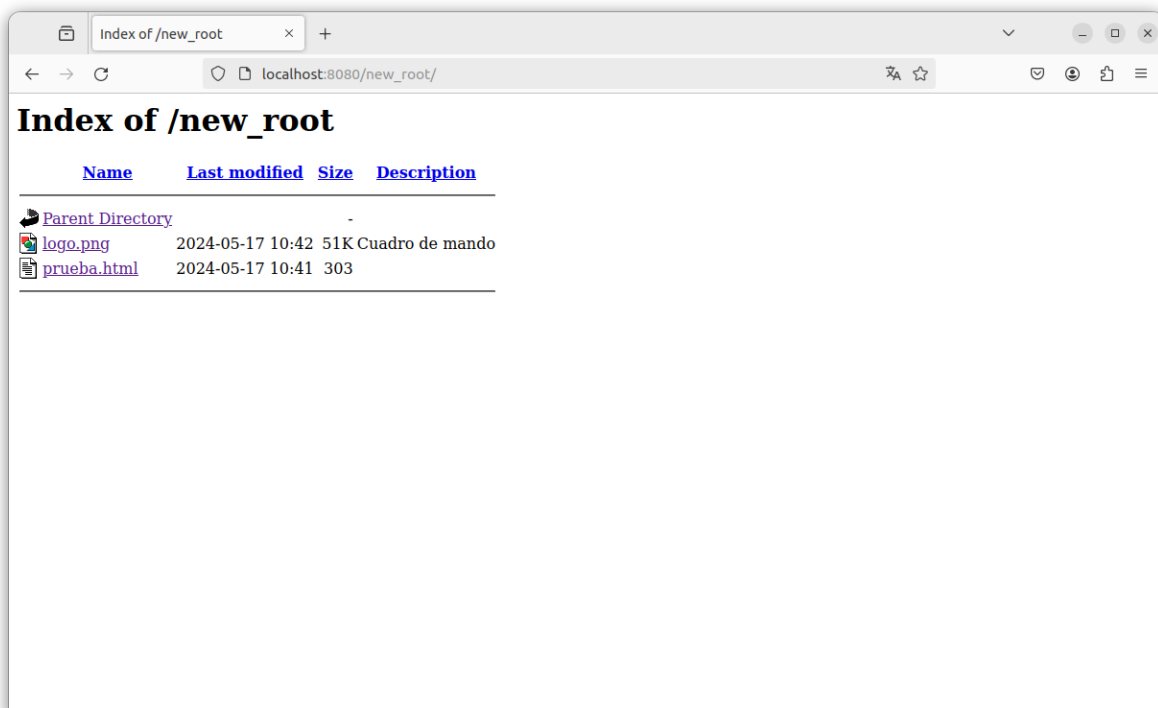


```
GNU nano 6.2 httpd.conf
#
    Require all granted
</Directory>

<Directory "/home/usuario/httpd/htdocs/new_root">
    Options Indexes FollowSymLinks
    AllowOverride All
    AddDescription "Cuadro de mando" logo.png
    Require all granted
</Directory>

#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
    DirectoryIndex index.html index.htm
</IfModule>
```

Y por último comprobamos que se muestra el index con la descripción correspondiente



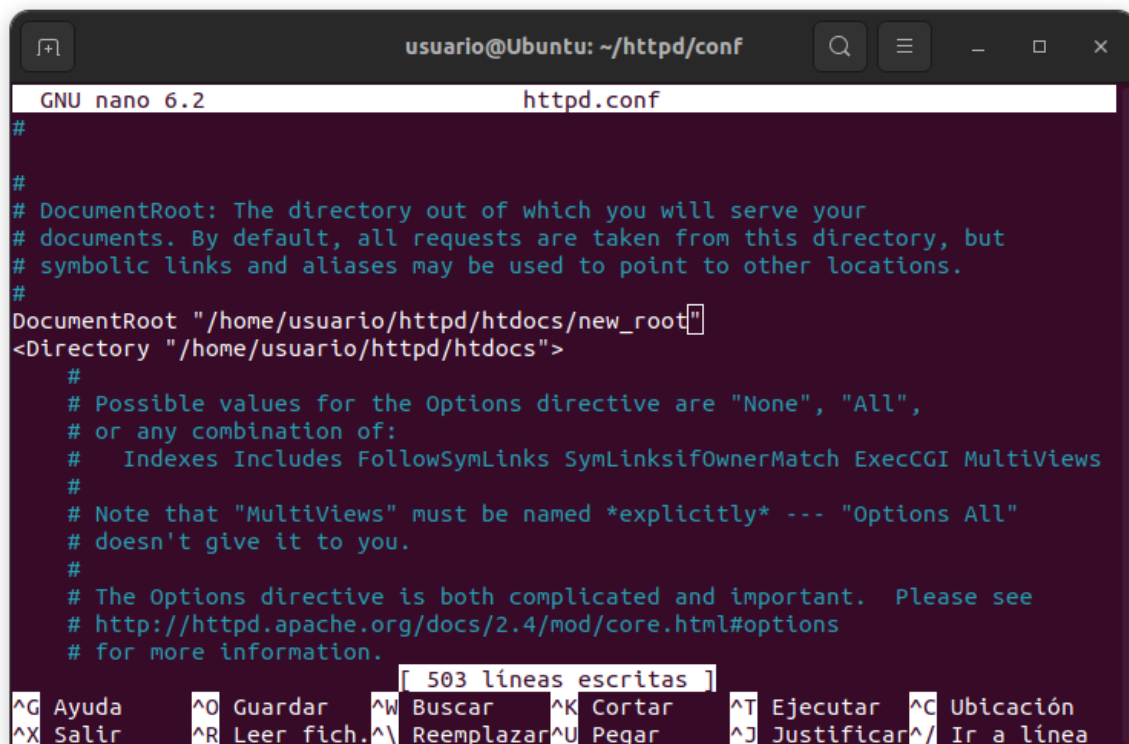
# Ejercicio 7

## Enunciado

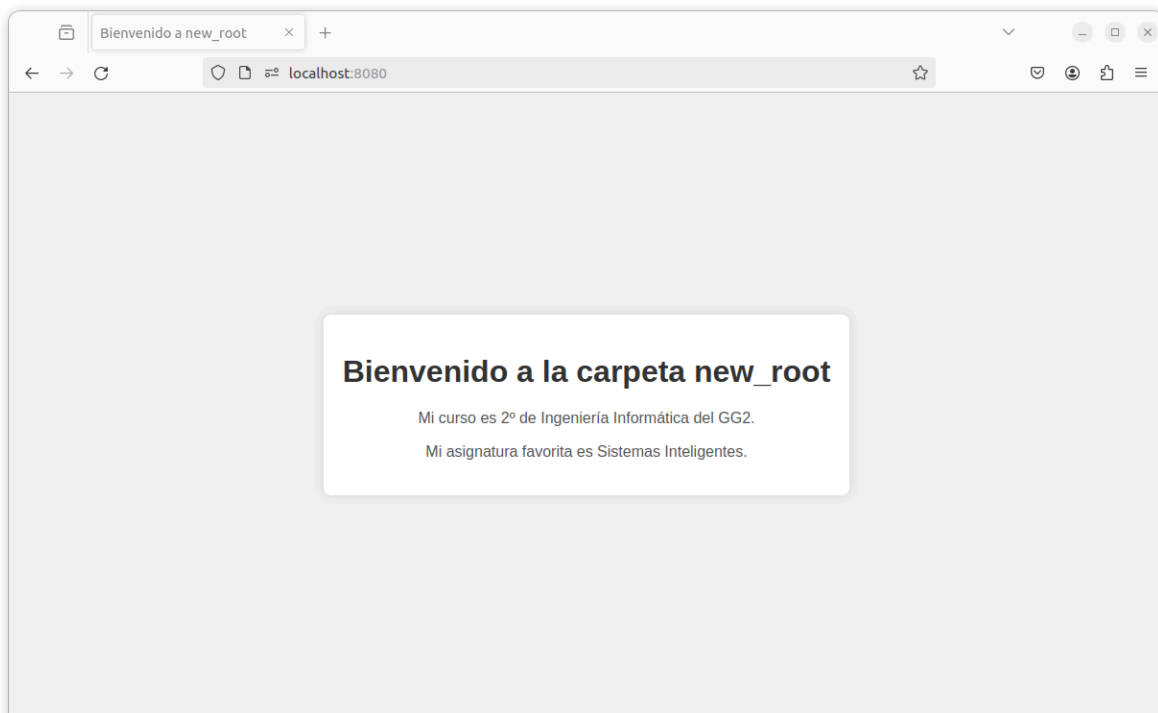
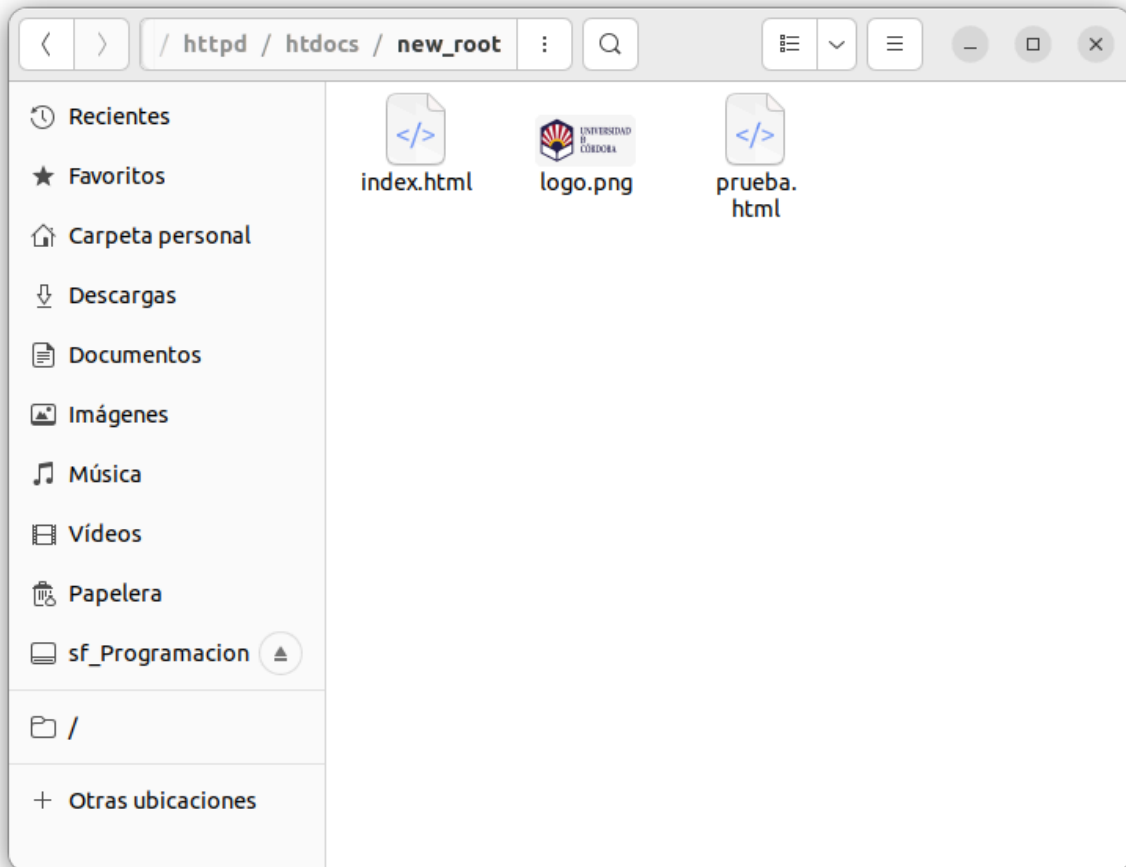
Los archivos que sirve Apache (los que puedo ver desde el cliente) están en una carpeta concreta. Cambia dicha carpeta (DocumentRoot), para que sea la carpeta creada antes nuevo\_root. Crear la carpeta y copiar dentro un conjunto de archivos .html para tu servidor web, dándole como nombre a uno de ellos index.html e introduciendo en él tu curso y tu asignatura preferida. Puedes buscar en Internet archivos básicos html.

## Resolución

Para cambiar la carpeta principal del localhost hay que abrir el fichero httpd.conf e ir a la directiva “DocumentRoot” y hay que poner la dirección del directorio que queremos usar como página principal del localhost



```
usuario@Ubuntu: ~/httpd/conf
GNU nano 6.2 httpd.conf
#
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/home/usuario/httpd/htdocs/new_root"
<Directory "/home/usuario/httpd/htdocs">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
[ 503 líneas escritas ]
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^_ Reemplazar  ^U Pegar      ^J Justificar ^_ Ir a línea
```



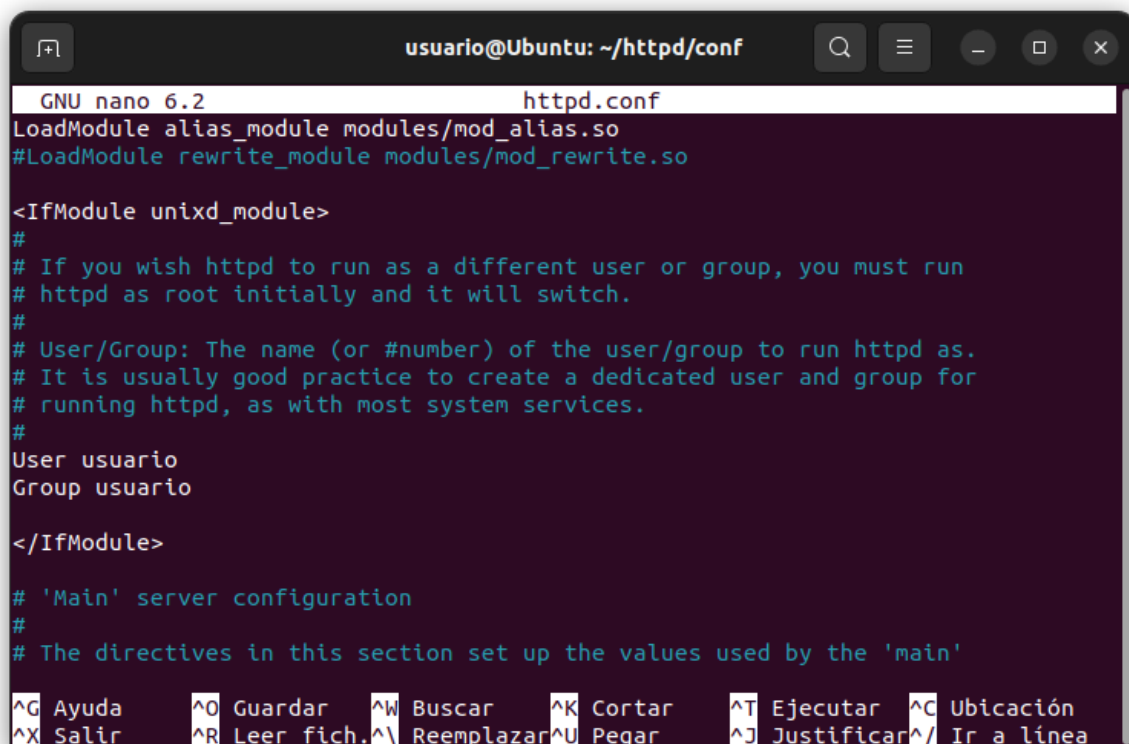
# Ejercicio 8

## Enunciado

Encuentra la directiva que especifica el usuario y grupo para la ejecución de Apache. ¿Cuáles son y para qué sirven?

## Resolución

Se encuentra dentro del fichero httpd.conf en las directivas “User” y “Group”



```
GNU nano 6.2 httpd.conf
LoadModule alias_module modules/mod_alias.so
#LoadModule rewrite_module modules/mod_rewrite.so

<IfModule unixd_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
User usuario
Group usuario

</IfModule>

# 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
```

<b>^G</b> Ayuda	<b>^O</b> Guardar	<b>^W</b> Buscar	<b>^K</b> Cortar	<b>^T</b> Ejecutar	<b>^C</b> Ubicación
<b>^X</b> Salir	<b>^R</b> Leer fich.	<b>^\\</b> Reemplazar	<b>^U</b> Pegar	<b>^J</b> Justificar	<b>^/</b> Ir a línea

La directiva “User” establece el nombre del usuario bajo el cual se ejecutará el servidor Apache y la directiva “Group” establece el grupo bajo el cual se ejecutará el servidor Apache

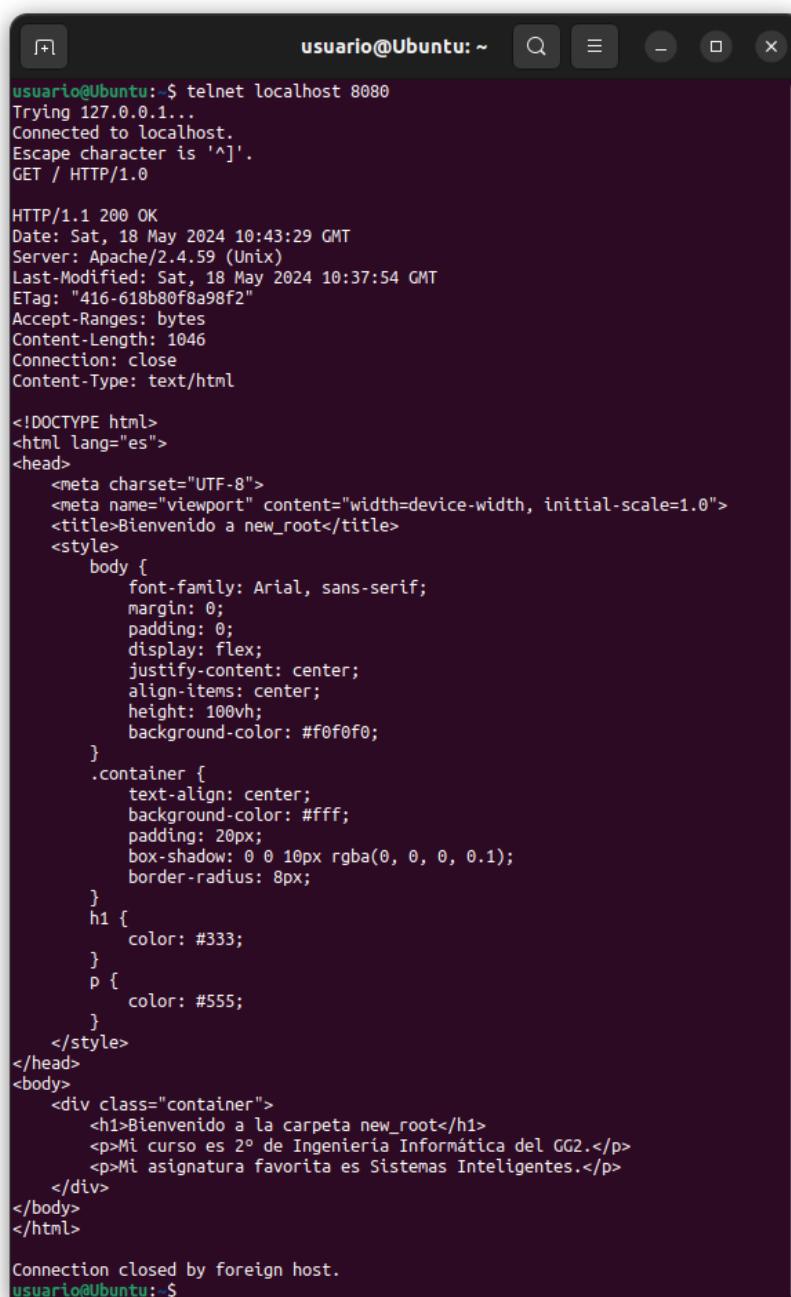
# Ejercicio 9

## Enunciado

Prueba a hacer un telnet al puerto del servidor (telnet localhost 80) y a mandarle mensajes HTML (escribe “GET / HTTP/1.0” y pulsa dos veces intro). Prueba a escribir HOLA y pulsar dos veces intro. Explica qué sucede y los códigos de error que devuelve el servidor.

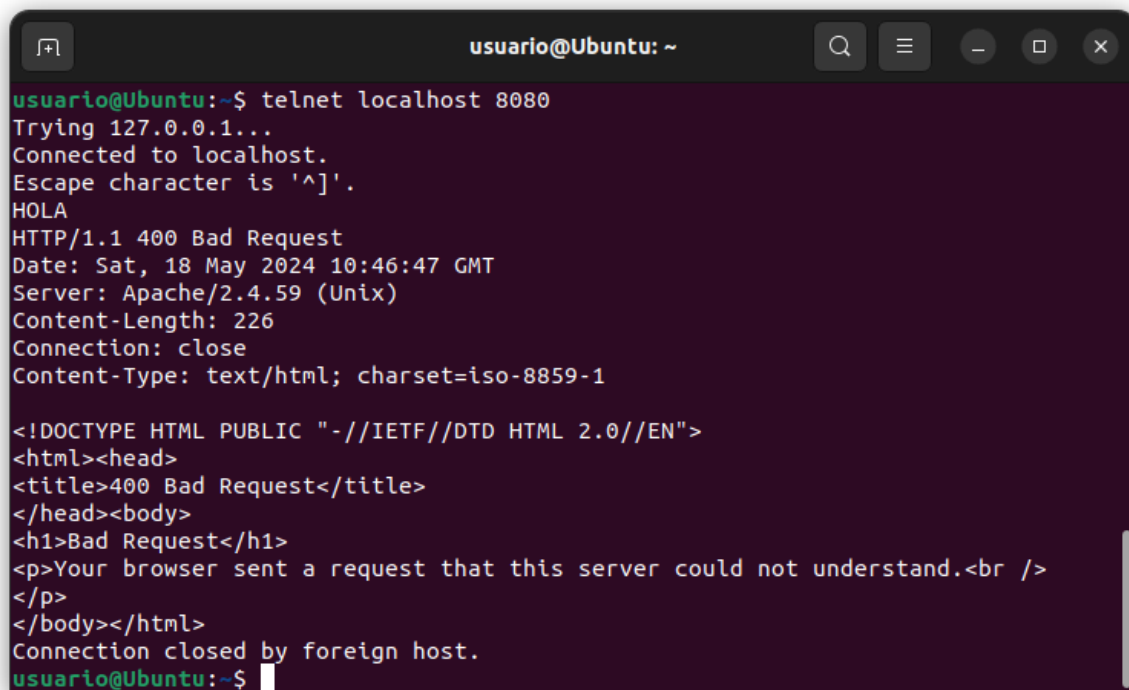
## Resolución

En primer lugar hacemos un telnet localhost 8080 y ponemos GET / HTTP/1.0:



```
usuario@Ubuntu: ~  
usuario@Ubuntu:~$ telnet localhost 8080  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^['.  
GET / HTTP/1.0  
  
HTTP/1.1 200 OK  
Date: Sat, 18 May 2024 10:43:29 GMT  
Server: Apache/2.4.59 (Unix)  
Last-Modified: Sat, 18 May 2024 10:37:54 GMT  
ETag: "416-618b80f8a98f2"  
Accept-Ranges: bytes  
Content-Length: 1046  
Connection: close  
Content-Type: text/html  
  
<!DOCTYPE html>  
<html lang="es">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Bienvenido a new_root</title>  
  <style>  
    body {  
      font-family: Arial, sans-serif;  
      margin: 0;  
      padding: 0;  
      display: flex;  
      justify-content: center;  
      align-items: center;  
      height: 100vh;  
      background-color: #f0f0f0;  
    }  
    .container {  
      text-align: center;  
      background-color: #fff;  
      padding: 20px;  
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
      border-radius: 8px;  
    }  
    h1 {  
      color: #333;  
    }  
    p {  
      color: #555;  
    }  
  </style>  
</head>  
<body>  
  <div class="container">  
    <h1>Bienvenido a la carpeta new_root</h1>  
    <p>Mi curso es 2º de Ingeniería Informática del GG2.</p>  
    <p>Mi asignatura favorita es Sistemas Inteligentes.</p>  
  </div>  
</body>  
</html>  
  
Connection closed by foreign host.  
usuario@Ubuntu:~$
```

A continuación vamos a hacer el telnet pero en este caso vamos a enviar un HOLA

A terminal window titled 'usuario@Ubuntu: ~' with standard window controls. The terminal shows the execution of 'telnet localhost 8080'. It connects to 127.0.0.1 and sends 'HOLA'. The server responds with an HTTP 400 Bad Request. The response body is an HTML document with the title '400 Bad Request' and a message: 'Your browser sent a request that this server could not understand.' The connection is then closed by the foreign host.

```
usuario@Ubuntu:~$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^['.
HOLA
HTTP/1.1 400 Bad Request
Date: Sat, 18 May 2024 10:46:47 GMT
Server: Apache/2.4.59 (Unix)
Content-Length: 226
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
</body></html>
Connection closed by foreign host.
usuario@Ubuntu:~$
```

Como podemos observar nos da el error 400 que significa que el el servidor no puede procesar o reconocer la solicitud, ya que el servidor no es capaz de procesar la operación HOLA.

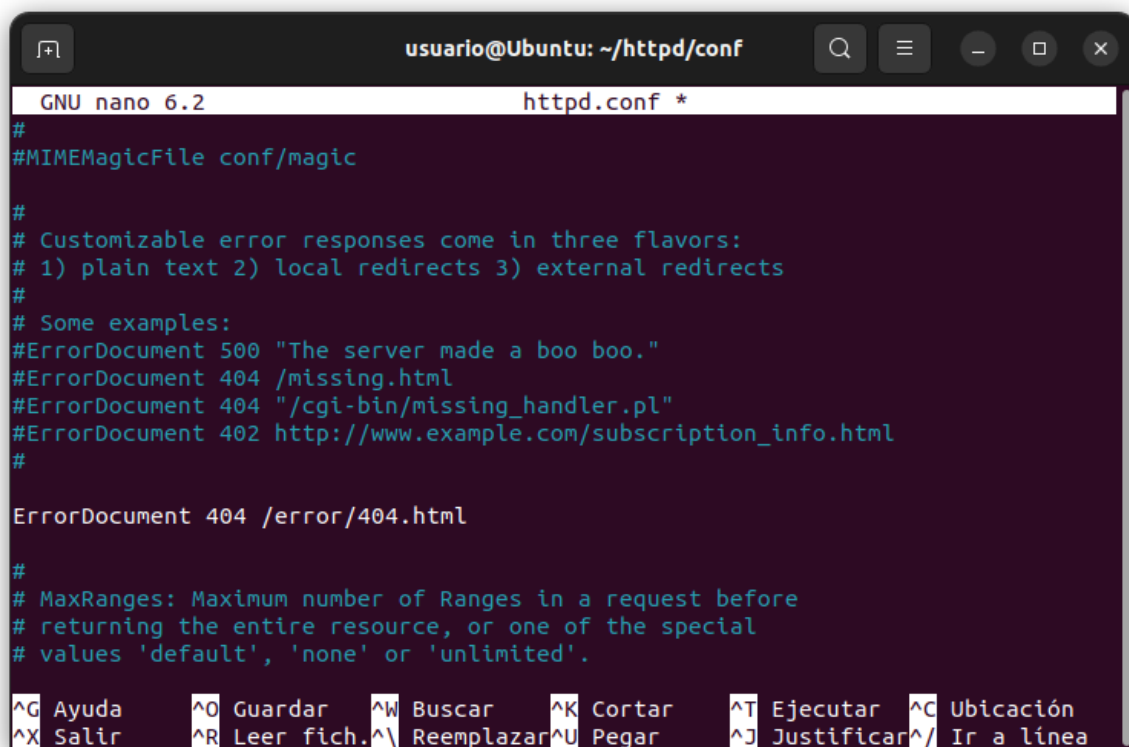
# Ejercicio 10

## Enunciado

Encuentra la directiva que se utiliza para la visualización de páginas de error y describe cada uno de los tipos de errores posibles. Modifica la directiva para mostrar una página HTML personalizada para el error 404. Busca en Internet algún diseño llamativo.

## Resolución

La directiva es “ErrorDocument”, en este caso he puesto “ErrorDocument 404 /error/404.html”, que quiere decir que si encuentra el error 404 va a mostrar el fichero 404.html que se encuentra en el directorio error

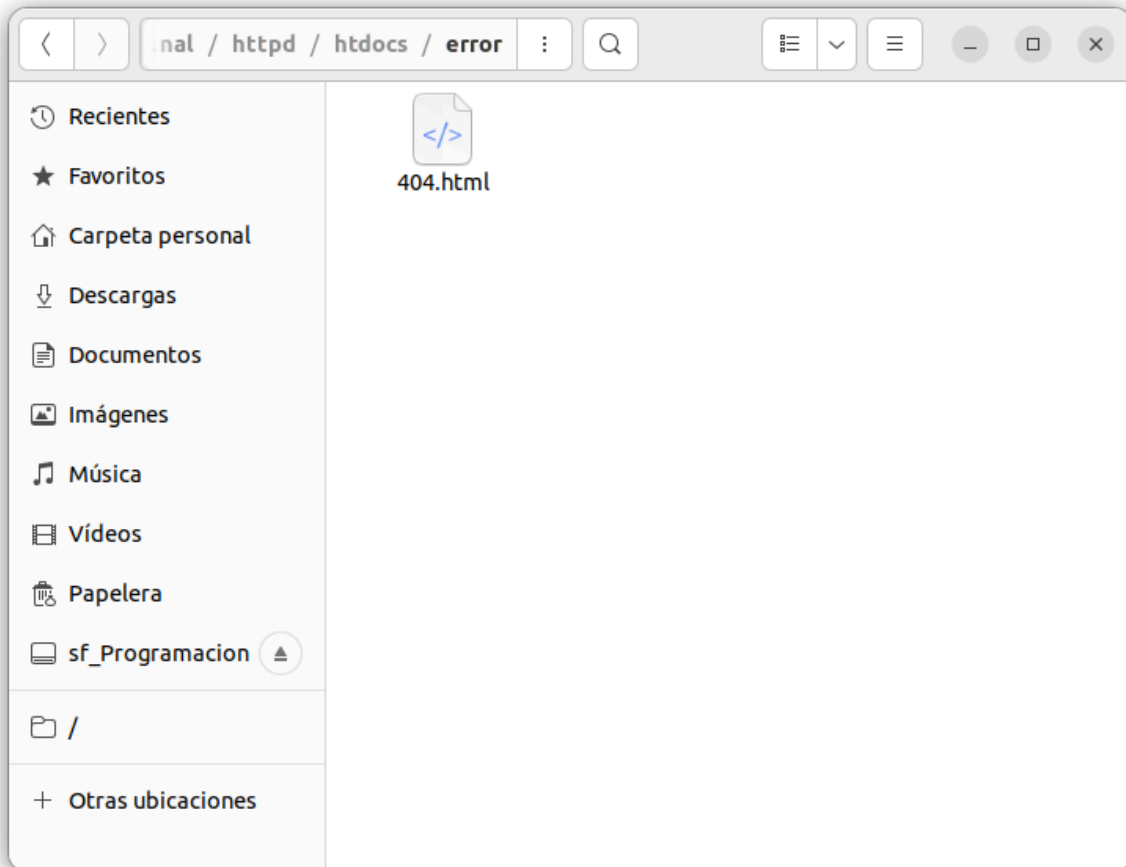


```
GNU nano 6.2 httpd.conf *
#
#MIMEMagicFile conf/magic
#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
ErrorDocument 404 /error/404.html
#
# MaxRanges: Maximum number of Ranges in a request before
# returning the entire resource, or one of the special
# values 'default', 'none' or 'unlimited'.
#
```

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación  
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar ^/ Ir a línea

Por lo tanto creamos una carpeta llamada error en la raíz de nuestra web y en esa carpeta introducimos el fichero 404.html





A continuación le he pedido el código HTML a Chat GPT y me ha dado el siguiente código, que lo introducimos en el fichero 404.html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Error 404 - Página no encontrada</title>
</style>
  body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
    background-color: #f0f0f0;
    font-family: 'Arial', sans-serif;
  }
  .container {
    text-align: center;
```

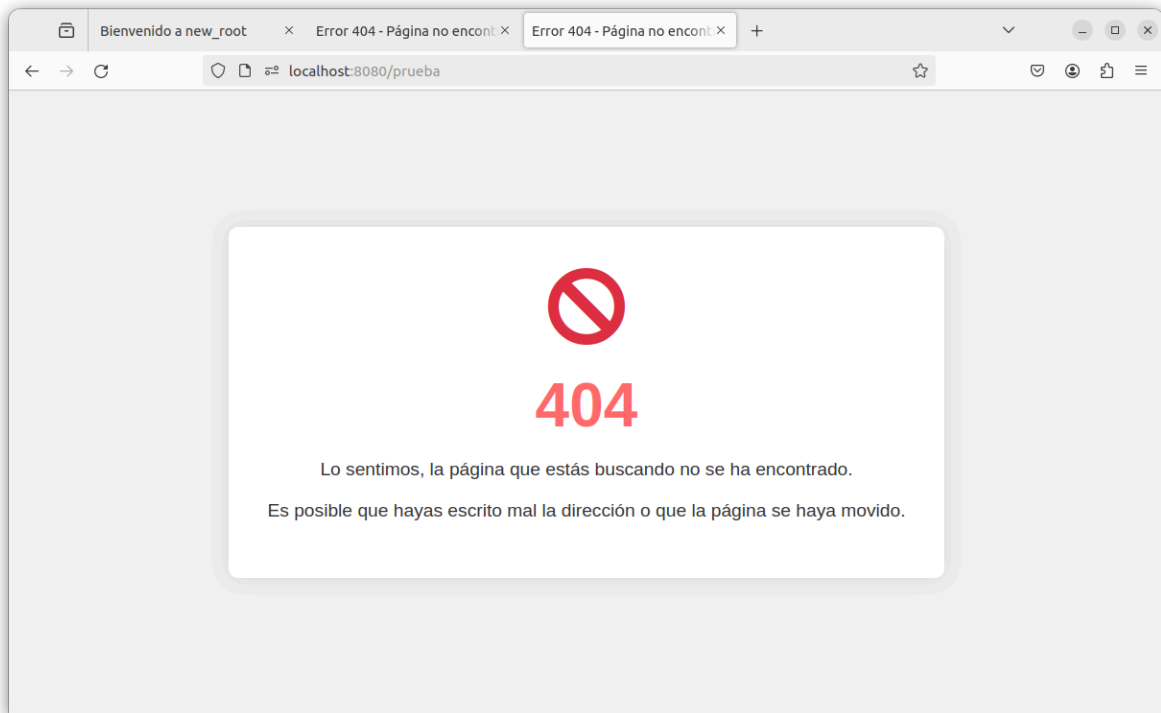
```

    background-color: #fff;
    padding: 40px;
    border-radius: 10px;
    box-shadow: 0 0 20px rgba(0,0,0,0.1);
    animation: fadeIn 1s ease-in-out;
  }
  @keyframes fadeIn {
    from { opacity: 0; }
    to { opacity: 1; }
  }
  h1 {
    font-size: 4em;
    color: #ff6b6b;
    margin: 0;
  }
  p {
    font-size: 1.2em;
    color: #333;
  }
  a {
    display: inline-block;
    margin-top: 20px;
    padding: 10px 20px;
    font-size: 1em;
    color: #fff;
    background-color: #ff6b6b;
    text-decoration: none;
    border-radius: 5px;
    transition: background-color 0.3s;
  }
  a:hover {
    background-color: #e65555;
  }
  .icon {
    font-size: 5em;
    color: #ff6b6b;
    margin-bottom: 20px;
  }
</style>
</head>
<body>
  <div class="container">
    <div class="icon"><img alt="Red circle with a slash" data-bbox="315 875 345 895"/></div>
    <h1>404</h1>
  </div>

```

```
<p>Lo sentimos, la página que estás buscando no se ha encontrado.</p>  
<p>Es posible que hayas escrito mal la dirección o que la página se haya movido.</p>  
</div>  
</body>  
</html>
```

Y como comprobamos se muestra la página siguiente al recibir el error 404:



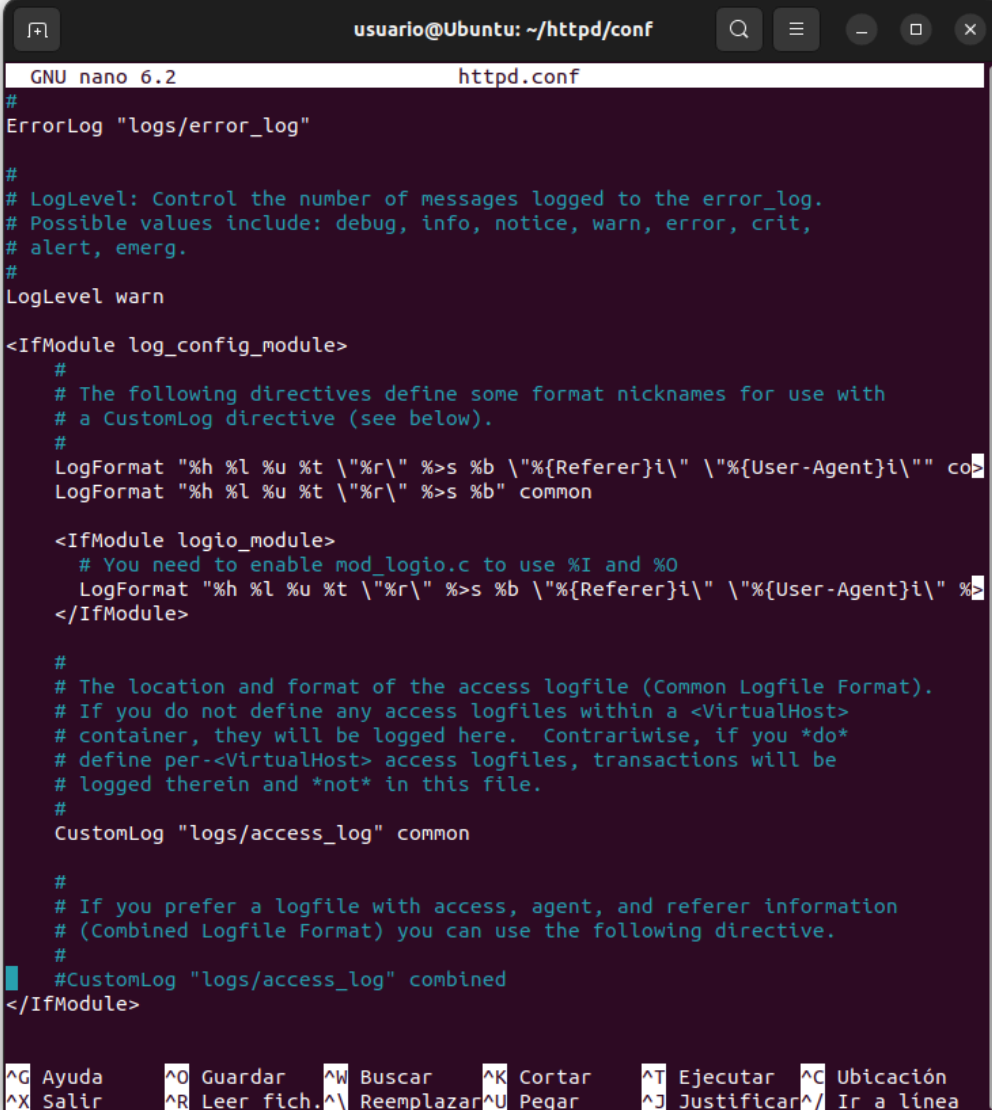
# Ejercicio 11

## Enunciado

Identifica las directivas relacionadas con los archivos de logs de Apache. Para ello desactiva las Options en un directorio del ejercicio 6, para que muestre el error 403 de Acceso Prohibido. Comprueba los logs y muestra las nuevas líneas que se han agregado. Reinicia apache y comprueba de nuevo si se ha incluido alguna línea y en qué archivos de log. Confeccione una tabla con los principales errores que se pueden dar en un servidor HTTP como Apache.

## Resolución

Las directivas relacionadas con los archivos logs son las siguientes: “ErrorLog”, “LogLevel”, “LogFormat” y “CustomLog”



The screenshot shows a terminal window titled 'usuario@Ubuntu: ~/httpd/conf' with a search bar and window controls. The editor is GNU nano 6.2, editing httpd.conf. The visible content is as follows:

```
#
ErrorLog "logs/error_log"

#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn

<IfModule log_config_module>
#
# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
#
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" co
LogFormat "%h %l %u %t \"%r\" %>s %b" common

<IfModule logio_module>
# You need to enable mod_logio.c to use %I and %O
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %>
</IfModule>

#
# The location and format of the access logfile (Common Logfile Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here. Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
#
CustomLog "logs/access_log" common

#
# If you prefer a logfile with access, agent, and referer information
# (Combined Logfile Format) you can use the following directive.
#
#CustomLog "logs/access_log" combined
</IfModule>
```

At the bottom, there is a status bar with keyboard shortcuts: ^G Ayuda, ^O Guardar, ^W Buscar, ^K Cortar, ^T Ejecutar, ^C Ubicación, ^X Salir, ^R Leer fich., ^\ Reemplazar, ^U Pegar, ^J Justificar, ^\_ Ir a línea.

Cuando prohibimos el acceso al directorio new\_root e intentamos acceder a él, en el fichero error\_log aparece lo siguiente:

```
[Sat May 18 13:25:24.006642 2024] [autoindex:error] [pid 8483:tid 135128193955392]  
[client 127.0.0.1:48510] AH01276: Cannot serve directory  
/home/usuario/httpd/htdocs/new_root/: No matching DirectoryIndex (index.html,index.htm)  
found, and server-generated directory index forbidden by Options directive
```

Y al reiniciar el servidor nos aparece lo siguiente en el fichero error\_log

```
[Sat May 18 13:27:39.746850 2024] [core:notice] [pid 3858:tid 135128306214720]  
AH00094: Command line: '/home/usuario/httpd/bin/httpd'
```

Los principales errores de un servidor HTTP como apache son los siguientes:

Error	Descripción
<u>400 Bad Request</u>	La solicitud del cliente no se puede procesar debido a una sintaxis incorrecta
<u>401 Unauthorized</u>	La solicitud requiere autenticación del usuario. El cliente debe autenticarse para obtener la respuesta solicitada
<u>403 Forbidden</u>	El servidor entiende la solicitud pero se niega a autorizarla
<u>404 Not Found</u>	El servidor no puede encontrar el recurso solicitado
<u>405 Method Not Allowed</u>	El método especificado en la solicitud no está permitido para el recurso solicitado
<u>408 Request Timeout</u>	El servidor no recibió una solicitud completa del cliente dentro del tiempo que estaba preparado para esperar
<u>500 Internal Server Error</u>	El servidor encontró una condición inesperada que le impidió cumplir con la solicitud
<u>502 Bad Gateway</u>	El servidor, actuando como una puerta de enlace, recibió una respuesta inválida del servidor de respaldo
<u>503 Service Unavailable</u>	El servidor no está disponible temporalmente para manejar la solicitud debido a mantenimiento o sobrecarga
<u>504 Gateway Timeout</u>	El servidor, actuando como una puerta de enlace, no recibió una respuesta a tiempo del servidor de respaldo
<u>505 HTTP Version Not Supported</u>	El servidor no admite la versión del protocolo HTTP utilizada en la solicitud

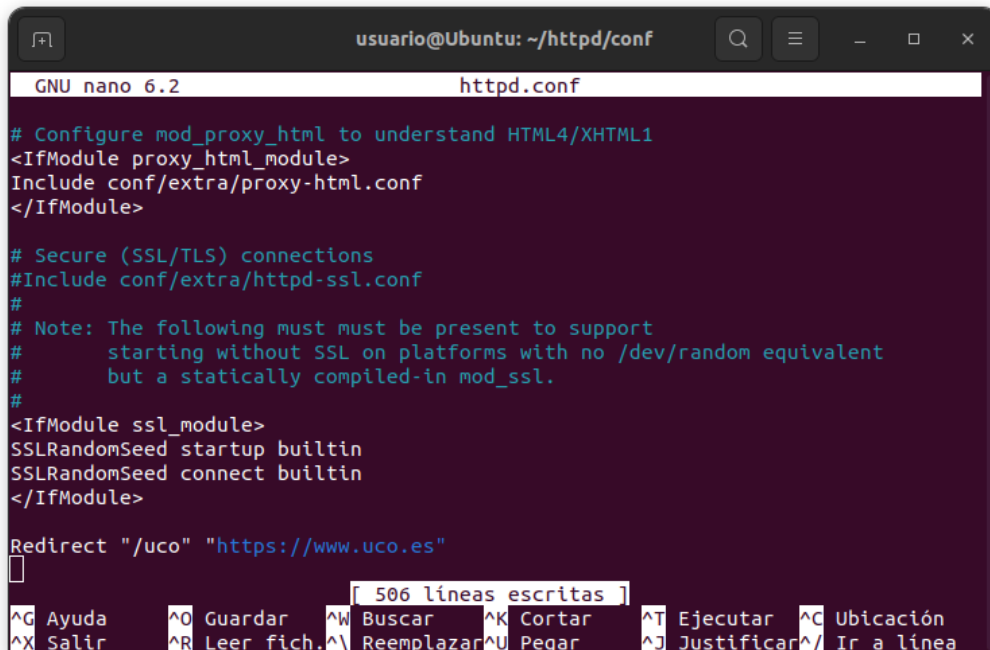
# Ejercicio 12

## Enunciado

Redirecciona la dirección localhost/uco a https://www.uco.es, de manera que al acceder a localhost/uco aparezca la web de la Universidad de Córdoba

## Resolución

Para ello añadimos lo siguiente al fichero httpd.conf “Redirect “/uco” “https://www.uco.es””



```
usuario@Ubuntu: ~/httpd/conf
GNU nano 6.2 httpd.conf

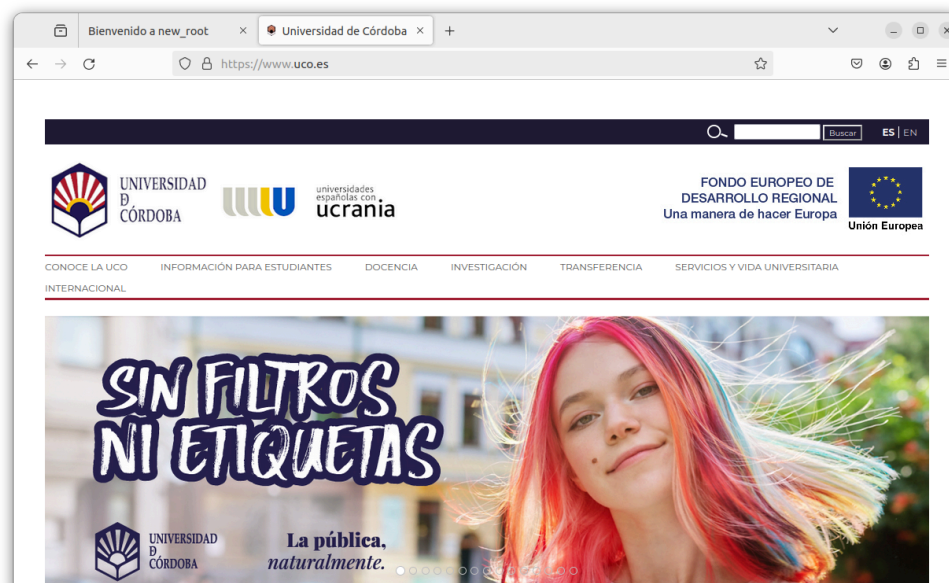
# Configure mod_proxy_html to understand HTML4/XHTML1
<IfModule proxy_html_module>
Include conf/extra/proxy-html.conf
</IfModule>

# Secure (SSL/TLS) connections
#Include conf/extra/httpd-ssl.conf
#
# Note: The following must must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>

Redirect "/uco" "https://www.uco.es"

```

Y comprobamos que nos redirecciona a la página web de la uco al poner localhost:8080/uco



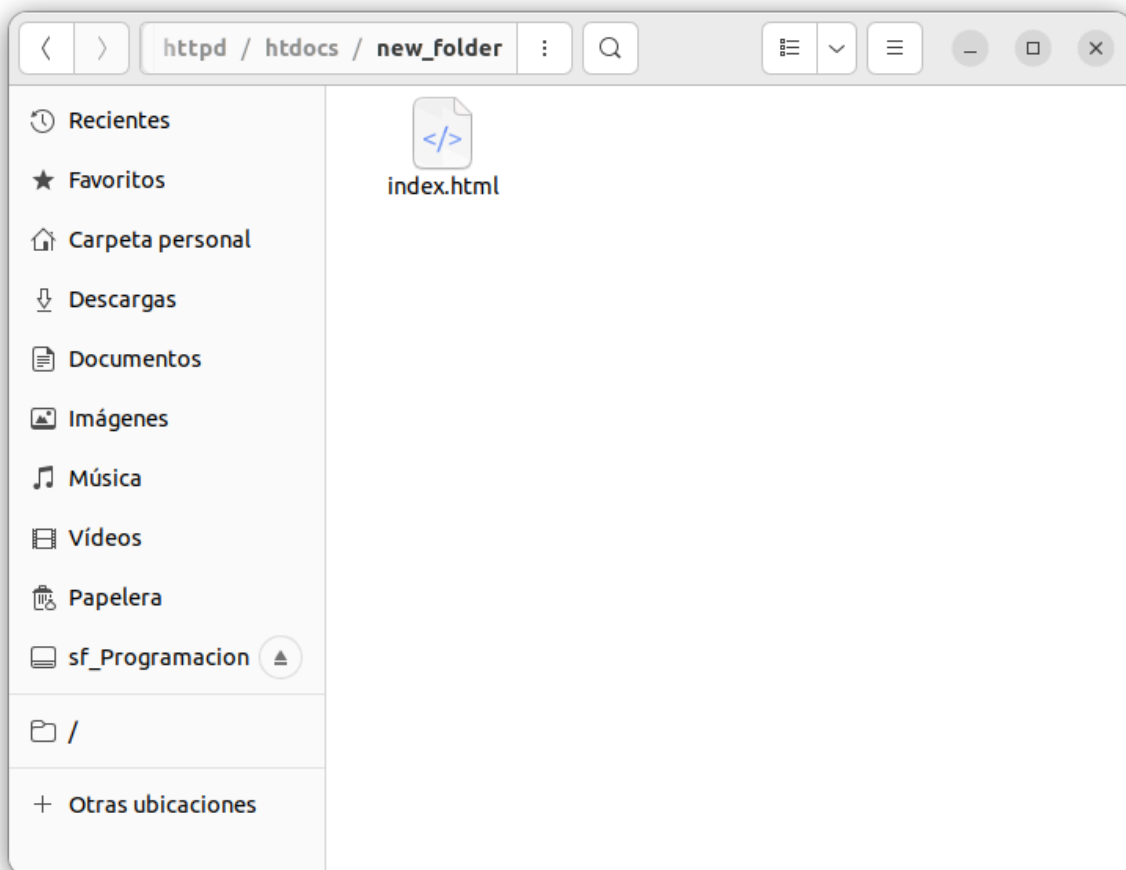
# Ejercicio 13

## Enunciado

Crea dos Hosts virtuales de manera que cuando un cliente se conecte al servidor usando localhost el servidor muestre la carpeta raíz original, y cuando se conecte usando miproyecto.local.com muestre el index.html de la subcarpeta /new\_folder.

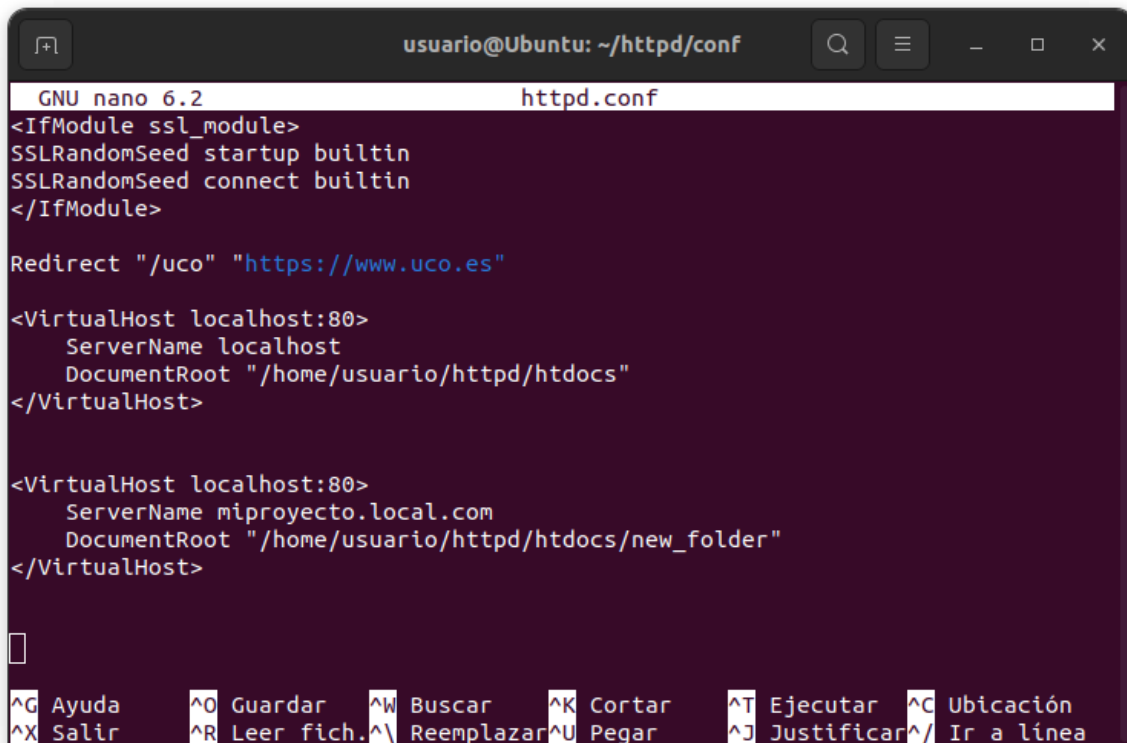
## Resolución

En primer lugar creamos la carpeta new\_folder y le añadimos un fichero index.html



A continuación en el fichero httpd.conf creamos dos directivas “VirtualHost” en el localhost:80. La primera con el nombre de localhost que nos lleva al directorio raíz de nuestra web de apache

Y otra con el nombre de miproyecto.local.com que nos lleva a la carpeta new\_folder que acabamos de crear



```
GNU nano 6.2 httpd.conf
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>

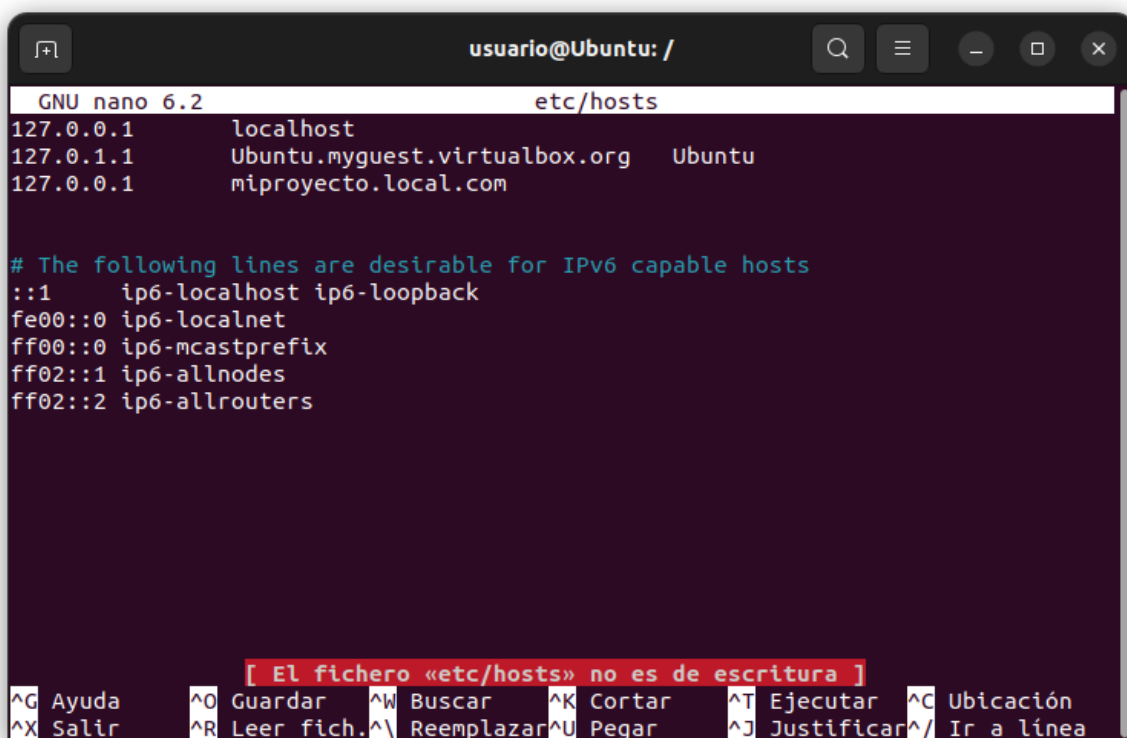
Redirect "/uco" "https://www.uco.es"

<VirtualHost localhost:80>
    ServerName localhost
    DocumentRoot "/home/usuario/httpd/htdocs"
</VirtualHost>

<VirtualHost localhost:80>
    ServerName miproyecto.local.com
    DocumentRoot "/home/usuario/httpd/htdocs/new_folder"
</VirtualHost>

^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^\ Reemplazar  ^U Pegar      ^J Justificar ^_ Ir a línea
```

Además antes de comprobar si esto funciona tenemos que irnos al fichero etc/hosts y añadir la línea 127.0.0.1 miproyecto.local.com para que al poner miproyecto.local.com en internet no lo busque en la red sino en el localhost



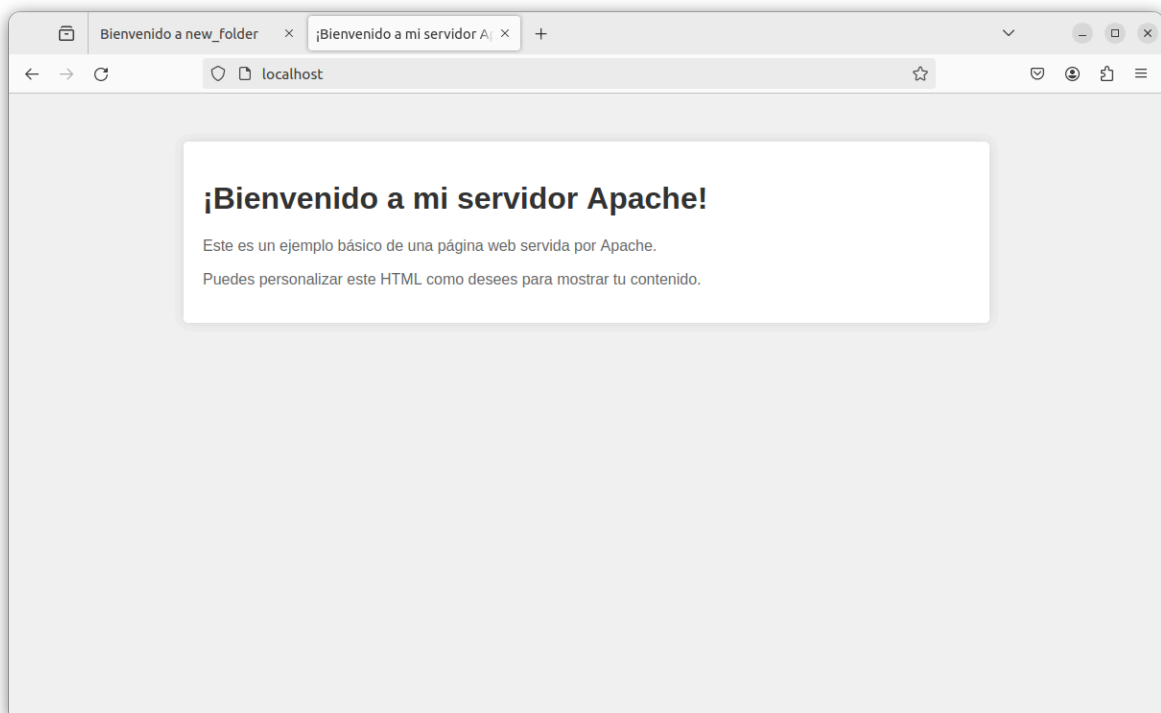
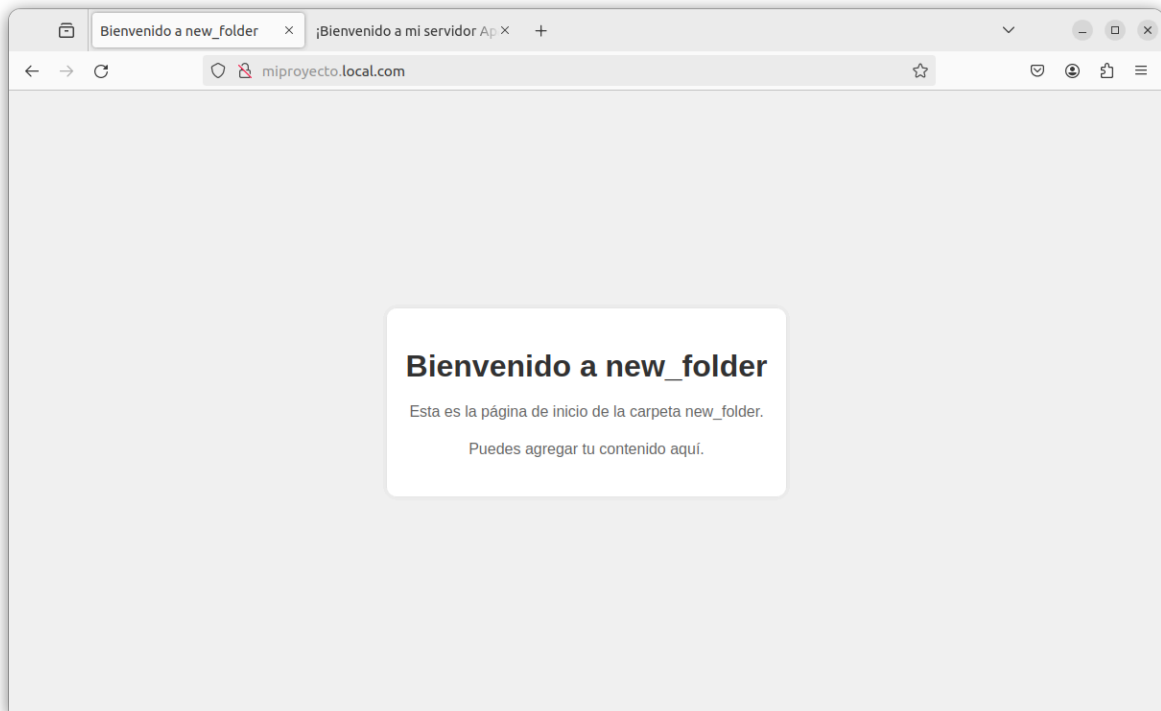
```
GNU nano 6.2 etc/hosts
127.0.0.1      localhost
127.0.1.1      Ubuntu.myquest.virtualbox.org  Ubuntu
127.0.0.1      miproyecto.local.com

# The following lines are desirable for IPv6 capable hosts
::1           ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters

[ El fichero «etc/hosts» no es de escritura ]
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^\ Reemplazar  ^U Pegar      ^J Justificar ^_ Ir a línea
```



A continuación comprobamos que al introducir ambas rutas en el navegador nos lleva a las páginas correspondientes



# Ejercicio 14

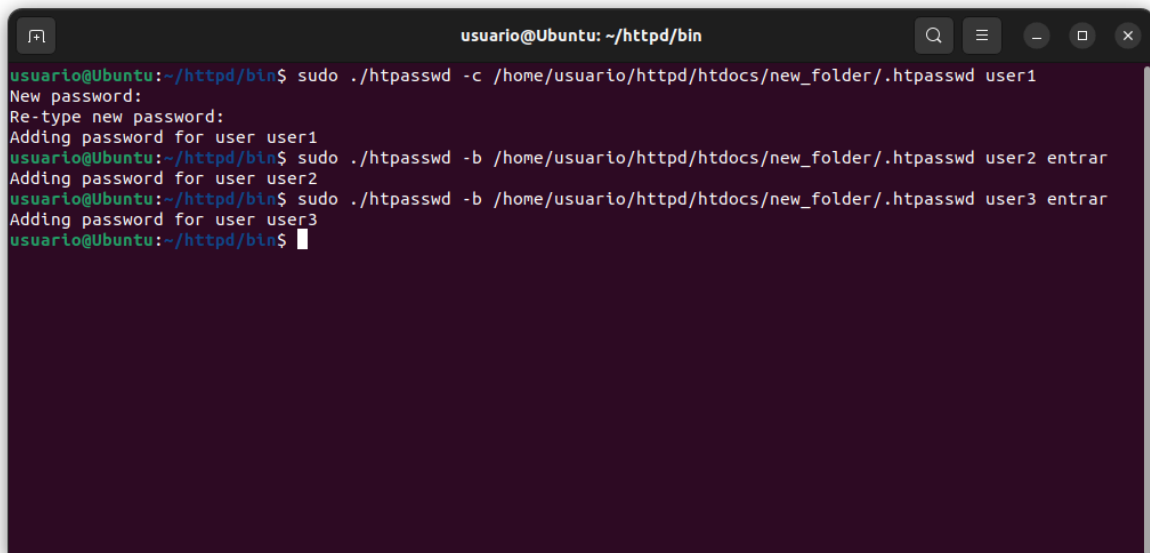
## Enunciado

Apache permite el acceso a recursos restringidos mediante la creación de usuarios y grupos que deberán autenticarse antes de acceder a dicho recurso protegido. Se debe:

- crear los usuarios y contraseñas correspondientes dentro de un archivo llamado `.htpasswd` (para esto, tendrás que utilizar el comando `./htpasswd -c .htpasswd usuario`, incluido en la carpeta `bin` de Apache, una vez por cada usuario a añadir y luego copiar el archivo resultante a un sitio protegido; ojo, el `-c` indica que el fichero se cree nuevo, por lo que sólo debe utilizarse para el primer usuario, los demás usuarios con `-b`)
- Aplicar las restricciones de acceso anteriores a la carpeta `new_folder` del host virtual `miproyecto.local.com`. Los usuarios que tendrán acceso a ese directorio serán: `user1`, `user2` y `user3`; y la contraseña, para todos será: `entrar`. Explica los pasos realizados para conseguirlo, y el contenido de los archivos creados.

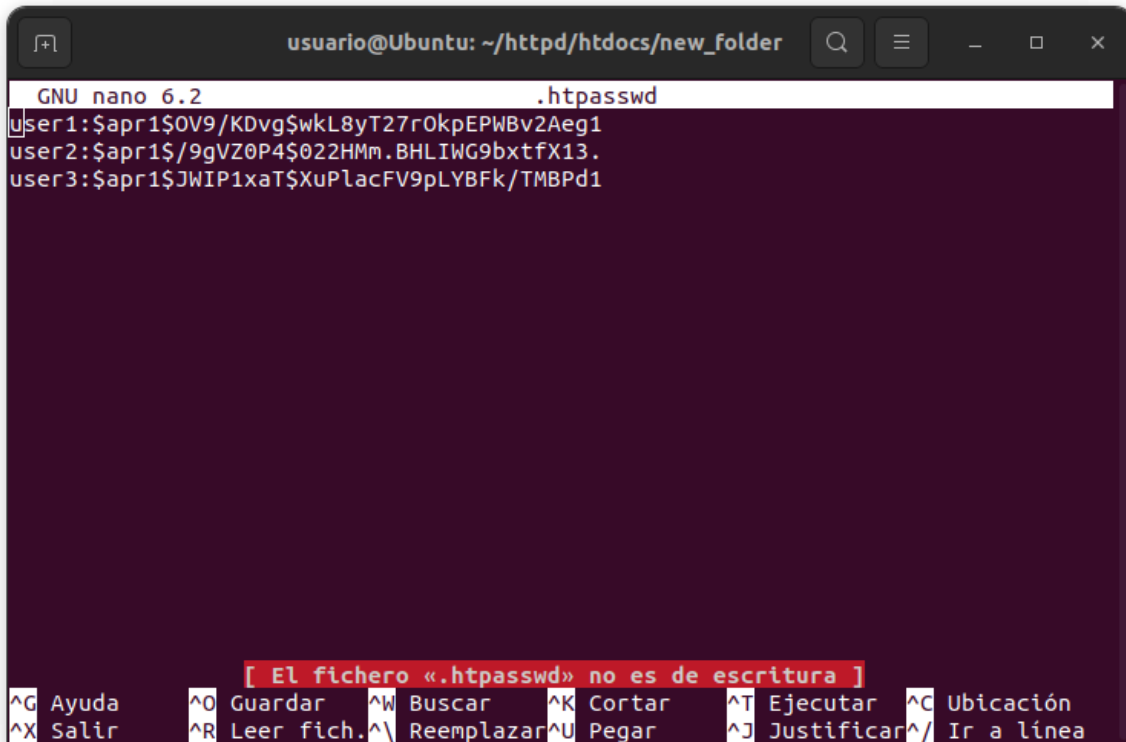
## Resolución

En primer lugar creamos los usuarios, para ello ejecutamos `htpasswd` que se encuentra en el directorio `bin`, el primer usuario lo creamos con la opción `-c` para crear el fichero `.htpasswd`, en la carpeta `new_folder` y los demás los creamos con la opción `-b`



```
usuario@Ubuntu: ~/httpd/bin
usuario@Ubuntu:~/httpd/bin$ sudo ./htpasswd -c /home/usuario/httpd/htdocs/new_folder/.htpasswd user1
New password:
Re-type new password:
Adding password for user user1
usuario@Ubuntu:~/httpd/bin$ sudo ./htpasswd -b /home/usuario/httpd/htdocs/new_folder/.htpasswd user2 entrar
Adding password for user user2
usuario@Ubuntu:~/httpd/bin$ sudo ./htpasswd -b /home/usuario/httpd/htdocs/new_folder/.htpasswd user3 entrar
Adding password for user user3
usuario@Ubuntu:~/httpd/bin$
```

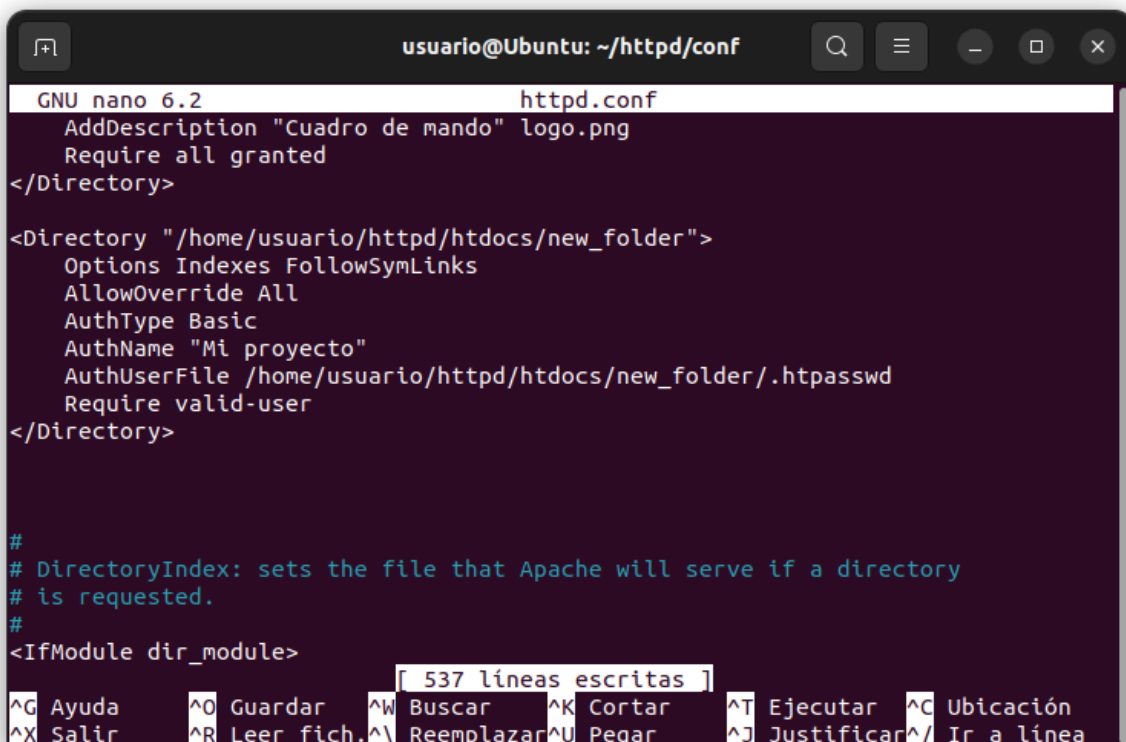
A continuación comprobamos que se han creado correctamente, para ello entramos en el fichero `.htpasswd` y vemos que tenemos los tres usuarios creados y sus contraseñas están encriptadas



```
usuario@Ubuntu: ~/httpd/htdocs/new_folder
GNU nano 6.2 .htpasswd
user1:$apr1$0V9/KDvg$wKl8yT27r0kpEPwBv2Aeg1
user2:$apr1$/9gVZ0P4$022HmM.BHLIWG9bxtfX13.
user3:$apr1$JWIP1xaT$XuPlacFV9pLYBFk/TMBPd1

[ El fichero «.htpasswd» no es de escritura ]
^G Ayuda  ^O Guardar  ^W Buscar  ^K Cortar  ^T Ejecutar  ^C Ubicación
^X Salir  ^R Leer fich.^_ Reemplazar^U Pegar  ^J Justificar^_/ Ir a línea
```

A continuación añadimos al fichero httpd.conf la directiva “Directory” con la ruta del directorio a new\_folder y dentro de esta configuramos “AuthType”, “AuthName”, “AuthUserFile” y “Require valid-user”



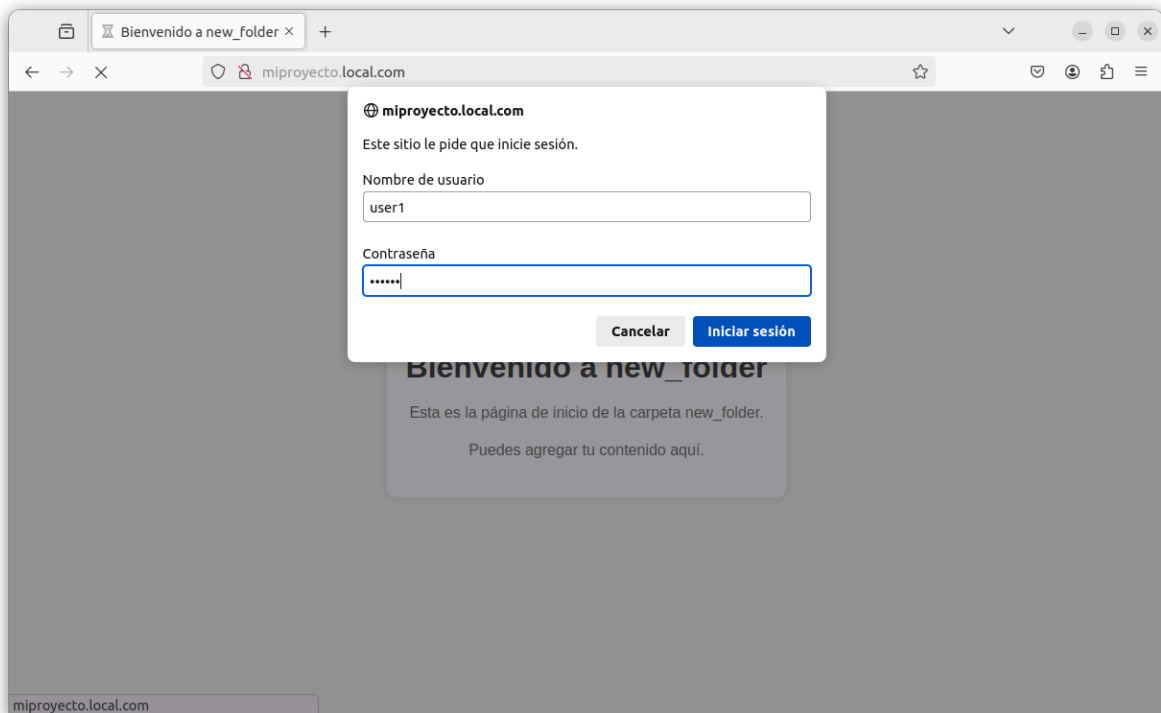
```
usuario@Ubuntu: ~/httpd/conf
GNU nano 6.2 httpd.conf
AddDescription "Cuadro de mando" logo.png
Require all granted
</Directory>

<Directory "/home/usuario/httpd/htdocs/new_folder">
    Options Indexes FollowSymLinks
    AllowOverride All
    AuthType Basic
    AuthName "Mi proyecto"
    AuthUserFile /home/usuario/httpd/htdocs/new_folder/.htpasswd
    Require valid-user
</Directory>

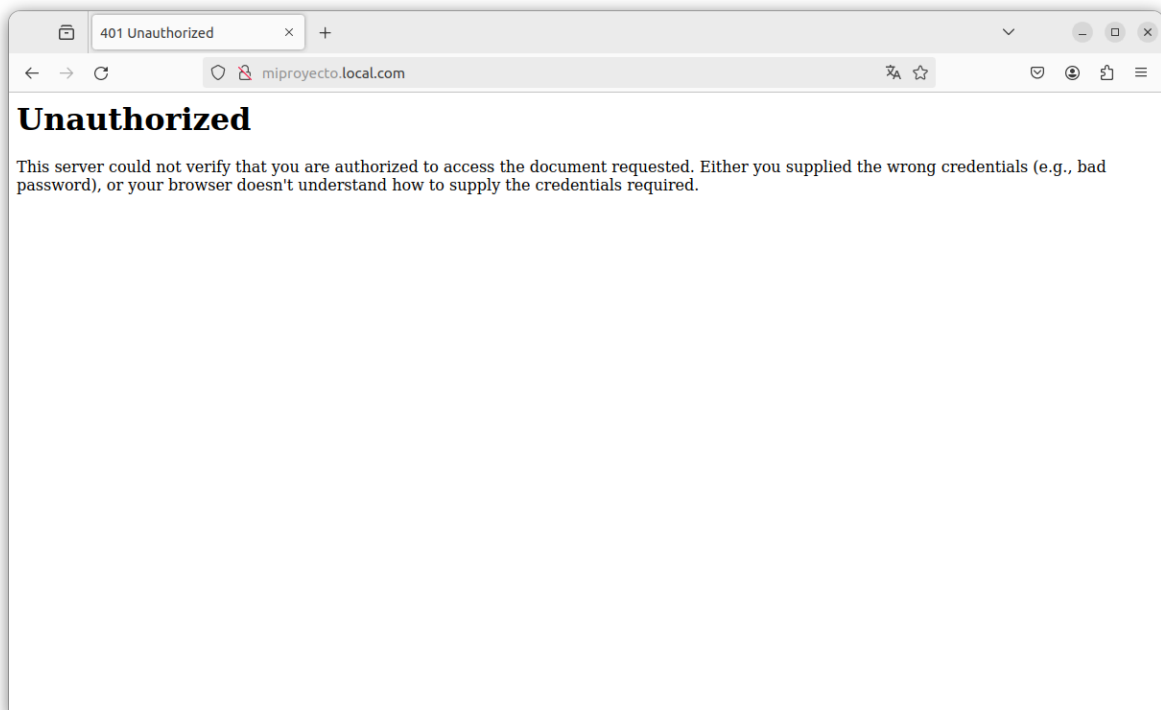
#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>

[ 537 líneas escritas ]
^G Ayuda  ^O Guardar  ^W Buscar  ^K Cortar  ^T Ejecutar  ^C Ubicación
^X Salir  ^R Leer fich.^_ Reemplazar^U Pegar  ^J Justificar^_/ Ir a línea
```

Comprobamos si nos pide un usuario para acceder



Y si cerramos la ventana emergente que nos pide el usuario y la contraseña vemos que nos da el error 401 que significa que no tenemos permiso para acceder porque no hemos autenticado el usuario



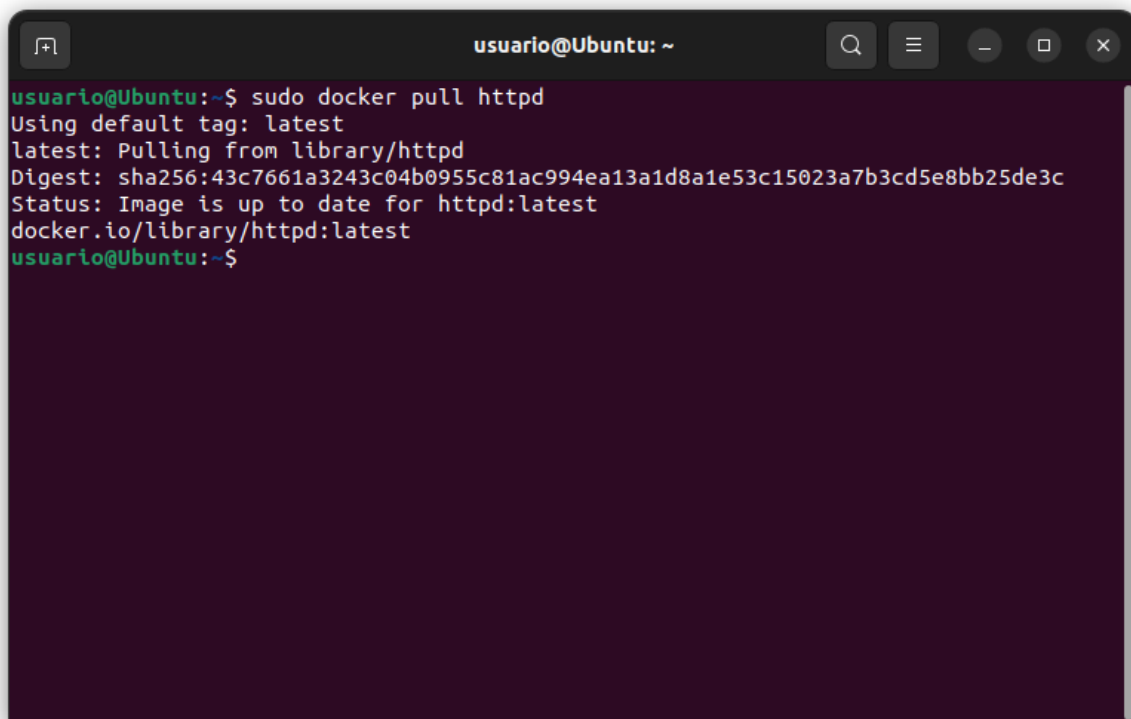
# Ejercicio 15

## Enunciado

Describir paso a paso la instalación de Apache en Ubuntu utilizando Docker y las ventajas que esto ofrece. Entregar capturas de pantalla del resultado.

## Resolución

En primer lugar tenemos que tener instalado docker, sino lo tenemos es tan sencillo como ejecutar el comando “sudo apt install docker”, a continuación ejecutamos el comando “sudo docker pull httpd”, para instalar el servidor de apache

A terminal window titled 'usuario@Ubuntu: ~' with standard window controls. The terminal shows the command 'sudo docker pull httpd' being executed. The output indicates that the 'latest' tag is used, the image is pulled from 'library/httpd', and its digest is 'sha256:43c7661a3243c04b0955c81ac994ea13a1d8a1e53c15023a7b3cd5e8bb25de3c'. The status is 'Image is up to date for httpd:latest' and the full path 'docker.io/library/httpd:latest' is shown. The prompt returns to 'usuario@Ubuntu:~\$'.

```
usuario@Ubuntu:~$ sudo docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
Digest: sha256:43c7661a3243c04b0955c81ac994ea13a1d8a1e53c15023a7b3cd5e8bb25de3c
Status: Image is up to date for httpd:latest
docker.io/library/httpd:latest
usuario@Ubuntu:~$
```

Para comprobar que se ha instalado correctamente ejecutamos el comando “sudo docker images” y comprobamos que nos aparece httpd

```
usuario@Ubuntu: ~  
usuario@Ubuntu:~$ sudo docker images  
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE  
odoo                 16.0       66b387113654  4 weeks ago   1.73GB  
docker-compose-flask_app  latest    5f6f5e6fc095  4 weeks ago   129MB  
hola-docker         latest    98e570c9bdd6  4 weeks ago   1.01GB  
httpd               latest    356125da0595  6 weeks ago   147MB  
postgres            13        6e7ee57f3efd  2 months ago  419MB  
redis               alpine     435993df2c8d  4 months ago  41MB  
hello-world         latest    d2c94e258dcb  12 months ago 13.3kB  
usuario@Ubuntu:~$
```

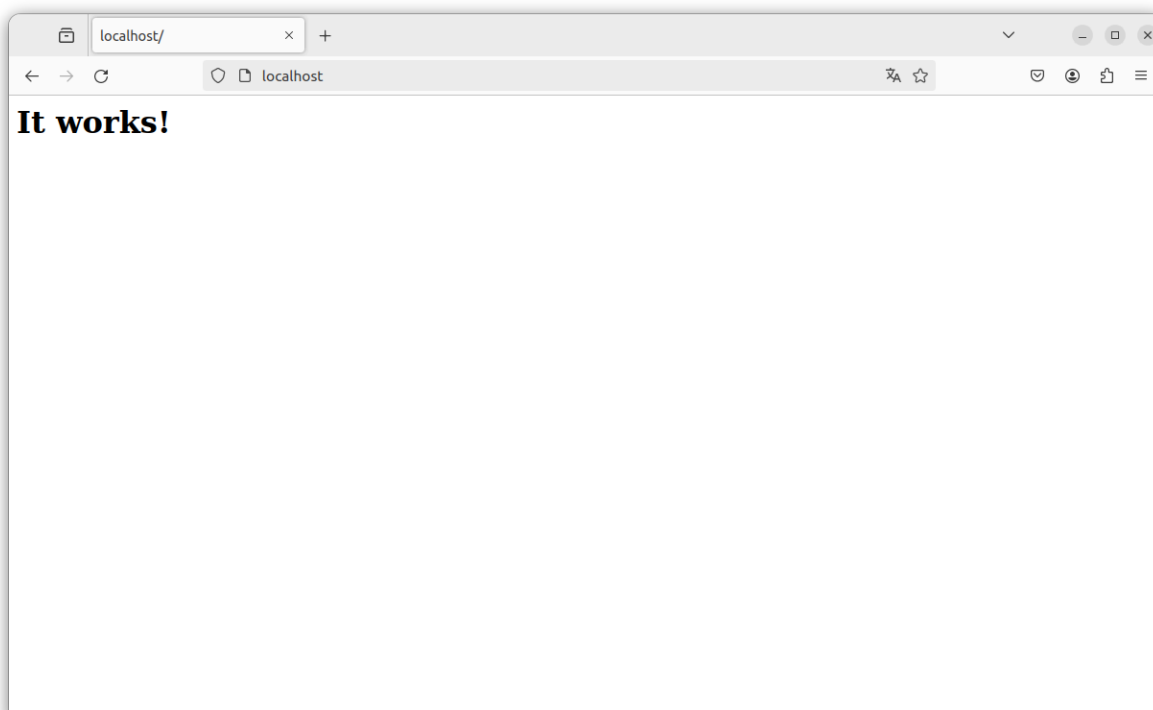
Para crear nuestro servidor de apache ejecutamos el siguiente comando “sudo docker run -d --name apache-server -p 80:80 httpd”

```
usuario@Ubuntu: ~  
usuario@Ubuntu:~$ sudo docker run -d --name apache-server -p 80:80 httpd  
3f3d79cb88a32bcb4c6269edbf847e979da3d68272f257fed471d4d4aaa309bb  
usuario@Ubuntu:~$
```

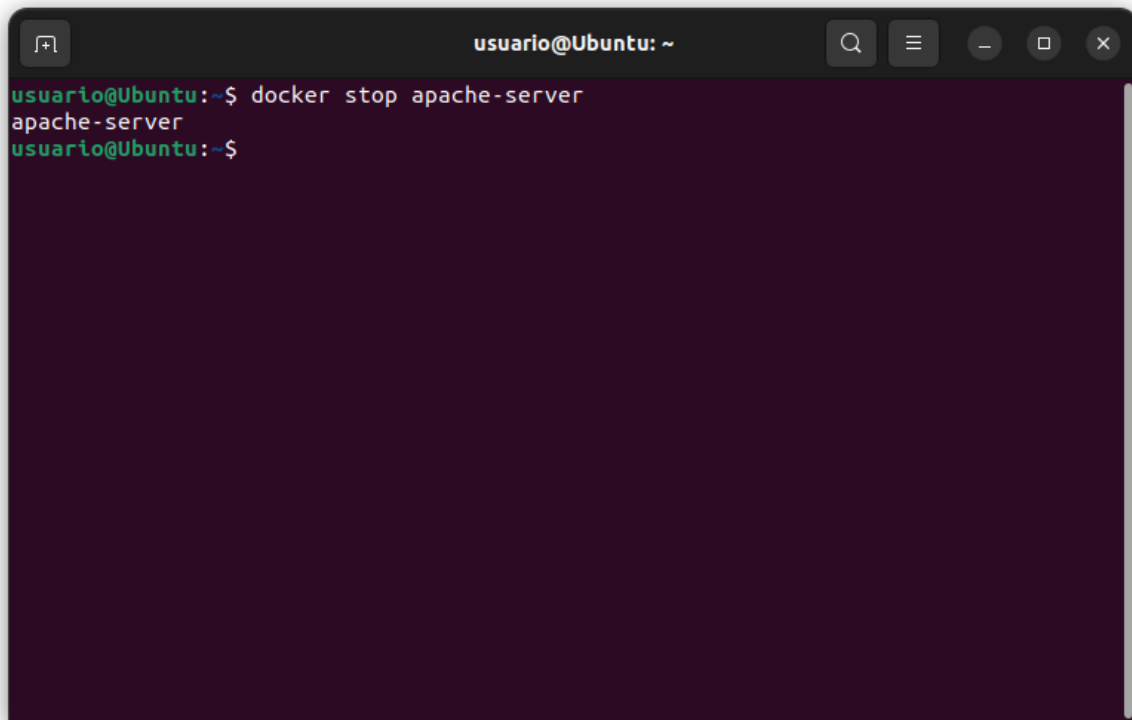
Para asegurarnos de que se está ejecutando ejecutamos el comando “docker ps” y comprobamos que está apache-server

```
usuario@Ubuntu: ~  
usuario@Ubuntu:~$ docker ps  
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS  
3f3d79cb88a3   httpd     "httpd-foreground"      16 seconds ago Up 14 seconds 0.0.0.0:80->80/tcp, :::80->80/tcp  
apache-server  
usuario@Ubuntu:~$
```

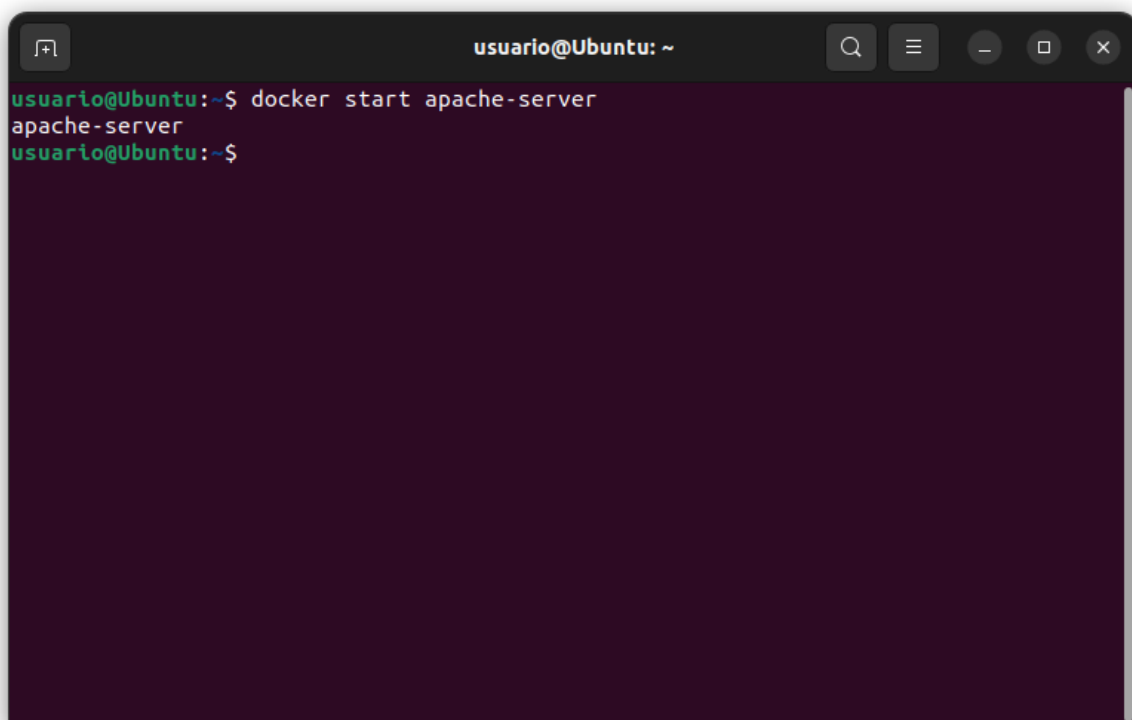
Y también nos metemos en el localhost para comprobar que se ha instalado y que está en funcionamiento correctamente



Una vez instalado utilizamos los comandos “docker stop apache-server” y “docker start apache-server” para apagar y encender nuestro servidor de apache

A terminal window titled 'usuario@Ubuntu: ~' with search, menu, and window control icons in the title bar. The terminal shows the command 'docker stop apache-server' being executed, followed by the output 'apache-server' on the next line, and the prompt 'usuario@Ubuntu:~\$' on the third line.

```
usuario@Ubuntu:~$ docker stop apache-server
apache-server
usuario@Ubuntu:~$
```

A terminal window titled 'usuario@Ubuntu: ~' with search, menu, and window control icons in the title bar. The terminal shows the command 'docker start apache-server' being executed, followed by the output 'apache-server' on the next line, and the prompt 'usuario@Ubuntu:~\$' on the third line.

```
usuario@Ubuntu:~$ docker start apache-server
apache-server
usuario@Ubuntu:~$
```

Usar Docker para instalar y gestionar un servidor Apache proporciona una forma moderna, eficiente y flexible de manejar tu infraestructura web. Sus principales ventajas son mayor portabilidad, aislamiento, facilidad de despliegue, escalabilidad, eficiencia de recursos, mantenimiento simplificado, seguridad y versatilidad que si instalamos nuestro servidor Apache de otra forma



# Ejercicio 16

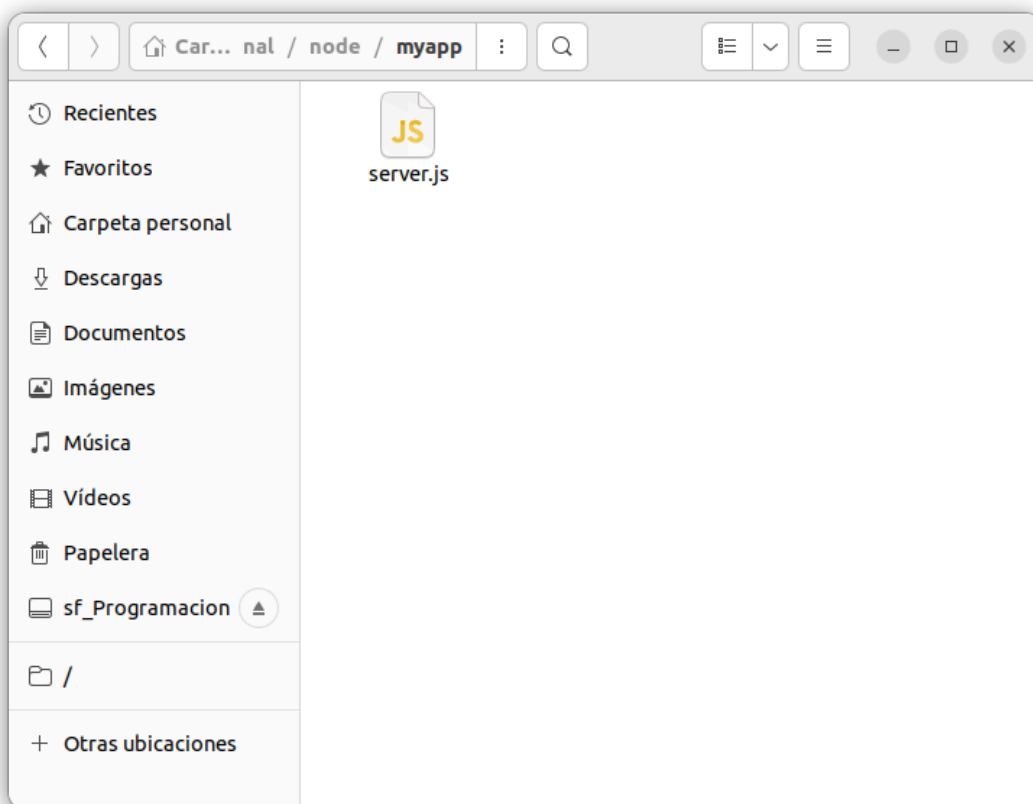
## Enunciado

(Opcional) ¿Qué es Nodejs? Instalar un servidor web basado en Nodejs utilizando los archivos de ejemplo subidos a Moodle. Haga una comparativa entre un servidor web Apache y un servidor web con Nodejs.

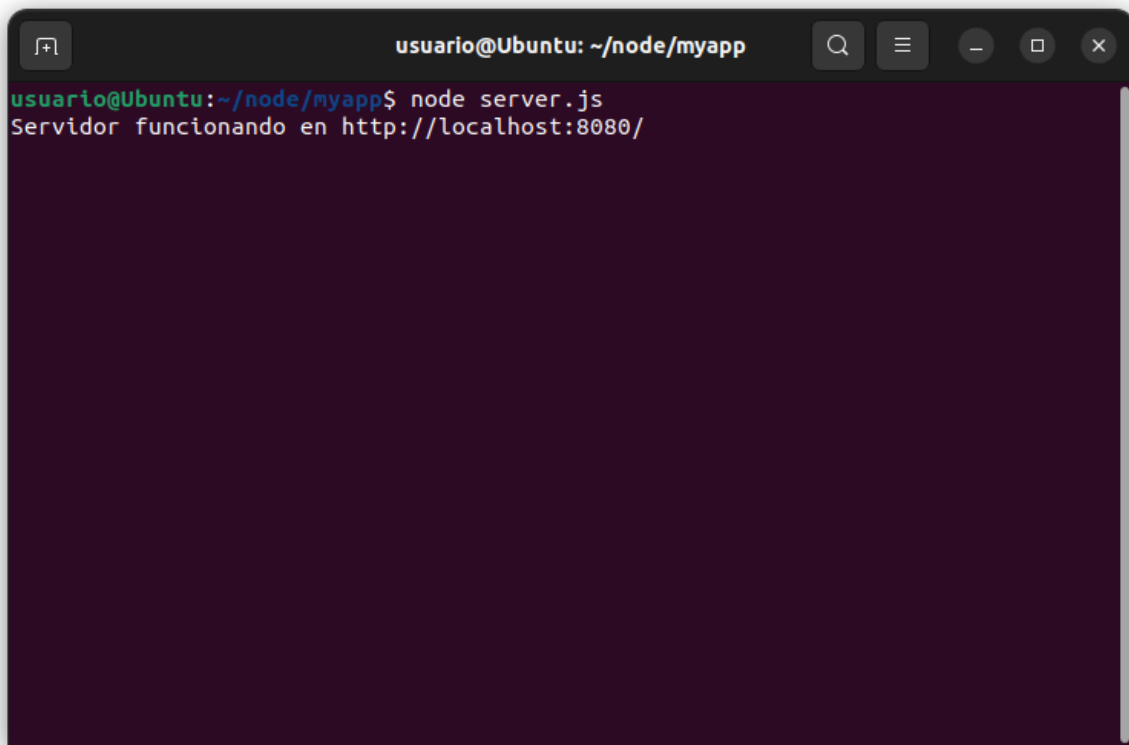
## Resolución

Nodejs es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google

Para instalar Nodejs utilizando los archivos de Moodle en primer lugar hay que descargarse el script para la instalación de Nodejs, y ejecutamos el script sin sudo para que se nos cree la carpeta node en nuestro usuario. A continuación creamos una carpeta dentro de la carpeta node y en esta nueva carpeta introducimos el fichero que se encuentra en moodle llamado “Archivo del servidor de Nodejs”

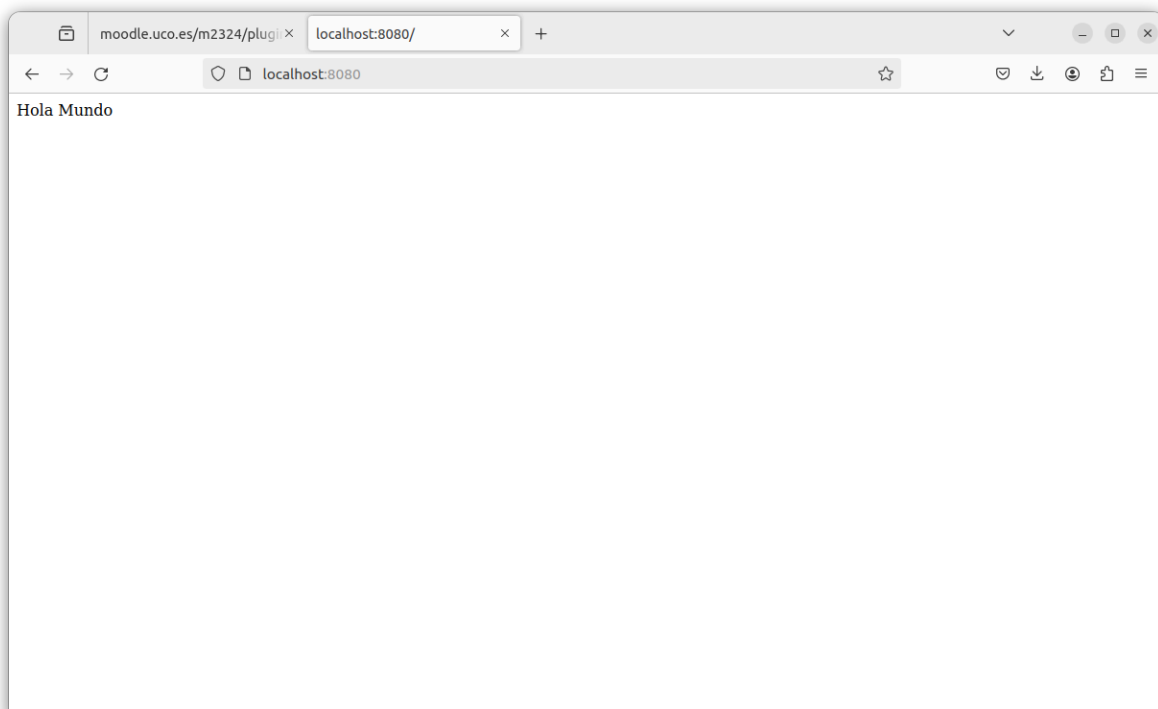


Y para ejecutar el servidor ponemos el comando “node server.js”



```
usuario@Ubuntu: ~/node/myapp
usuario@Ubuntu:~/node/myapp$ node server.js
Servidor funcionando en http://localhost:8080/
```

Y comprobamos que nuestro servidor está ejecutándose correctamente



Apache es más tradicional, con una sintaxis compleja y una amplia personalización. Nodejs es moderno, rápido y escalable, y se basa en JavaScript. Ambos tienen sus casos de uso específicos, y la elección depende de tus necesidades