

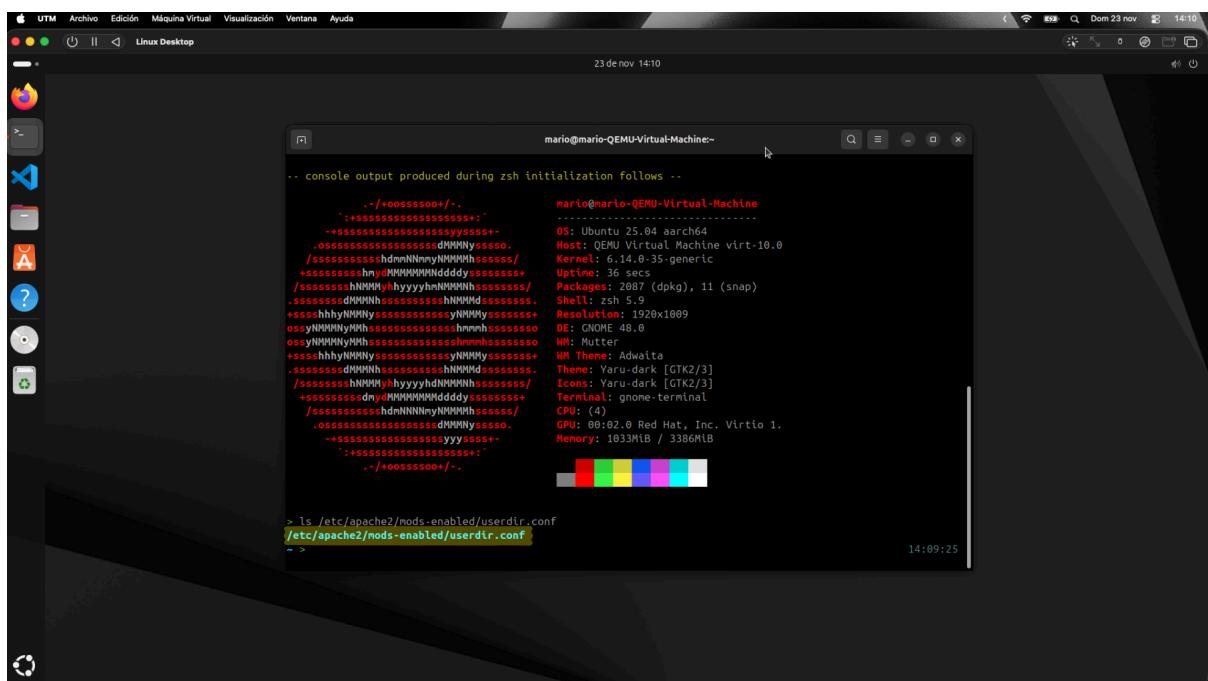
1. CONFIGURACIÓN WEB DE USUARIO (`mod_userdir`)

El objetivo es que cada usuario del sistema tenga su propia web accesible poniendo `localhost/~usuario`.

PASO 1.1: COMPROBAR Y ACTIVAR EL MÓDULO

La práctica pide comprobar si está habilitado.

- Comando: `ls /etc/apache2/mods-enabled/userdir.conf`
 - *Si no sale nada, no está habilitado.*



```
-- console output produced during zsh initialization follows --
mario@mario-QEMU-Virtual-Machine:~-----+
OS: Ubuntu 25.04 aarch64
Host: QEMU Virtual Machine virt-10.0
Kernel: 6.1.0-35-generic
Uptime: 36 sec
Packages: 2087 (dpkg), 11 (snap)
Shells: zsh 5.9
Resolution: 1920x1000
DE: GNOME 48.0
WM: Mutter
WM Theme: Adwaita
Theme: Yaru-dark [GTK2/3]
Icons: Yaru-dark [GTK2/3]
Terminal: gnome-terminal
CPU: (4)
GPU: 00:02.0 Red Hat, Inc. Virtio 1.
Memory: 1033MiB / 3386MiB
-----+
mario@mario-QEMU-Virtual-Machine:~
```

```
> ls /etc/apache2/mods-enabled/userdir.conf
/etc/apache2/mods-enabled/userdir.conf
->
```

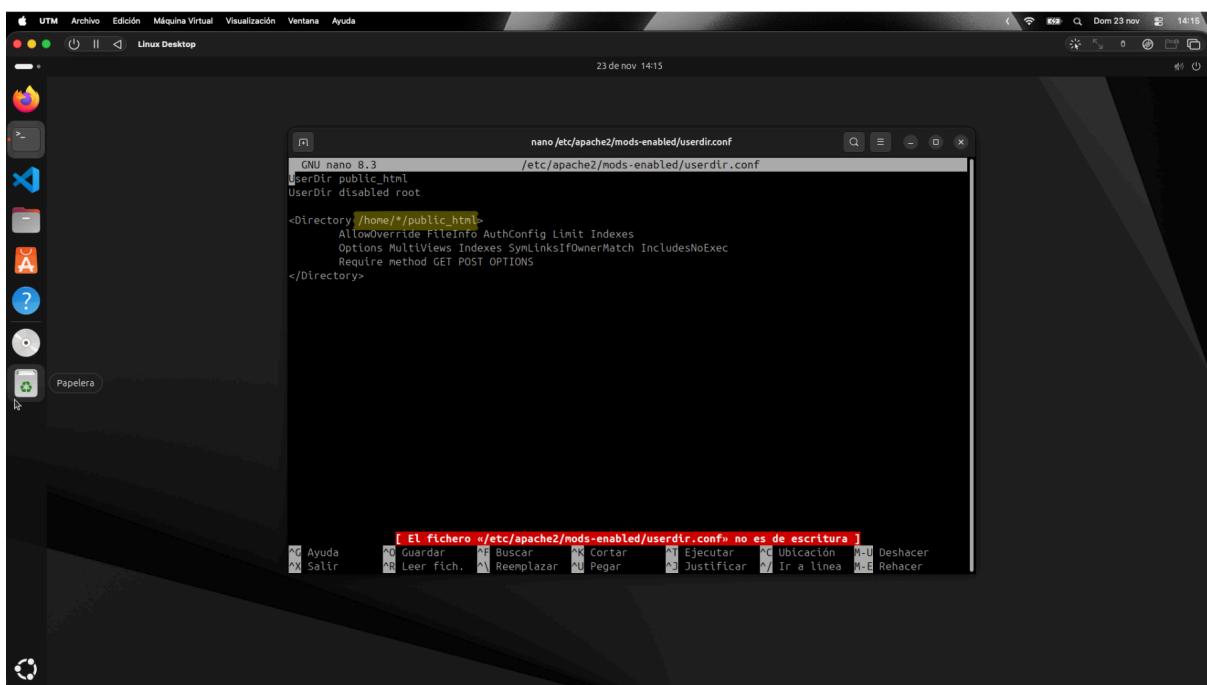
Acción (Habilitarlo):

```
sudo a2enmod userdir
```

```
sudo systemctl restart apache2
```

PASO 1.2: VERIFICAR CONFIGURACIÓN Y PREGUNTAS

- **Abre el fichero de configuración:** `nano /etc/apache2/mods-enabled/userdir.conf`.
- **Respuesta a la pregunta:** ¿Está habilitado para todos? Sí, normalmente pone `UserDir public_html`, lo que aplica a todos los usuarios válidos. Si ves `UserDir disabled root`, significa que está desactivado solo para root.
- **Respuesta a la pregunta:** ¿En qué directorio? En el directorio `public_html` dentro de la carpeta personal (`home`) de cada usuario.



PASO 1.3: CREAR LA PÁGINA PERSONAL

Ojo aquí: El módulo userdir busca por defecto la carpeta public_html, pero el enunciado dice "crea una página... en el directorio que corresponda". Vamos a usar el estándar public_html para asegurar que funcione.

Ir a tu carpeta personal

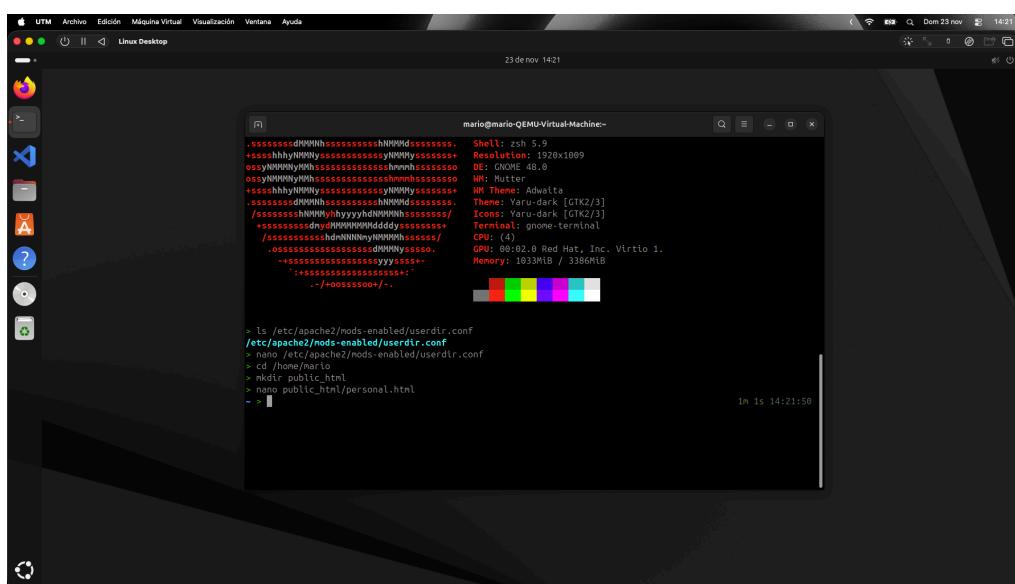
```
cd /home/mario/
```

Crear la carpeta pública (necesaria por defecto)

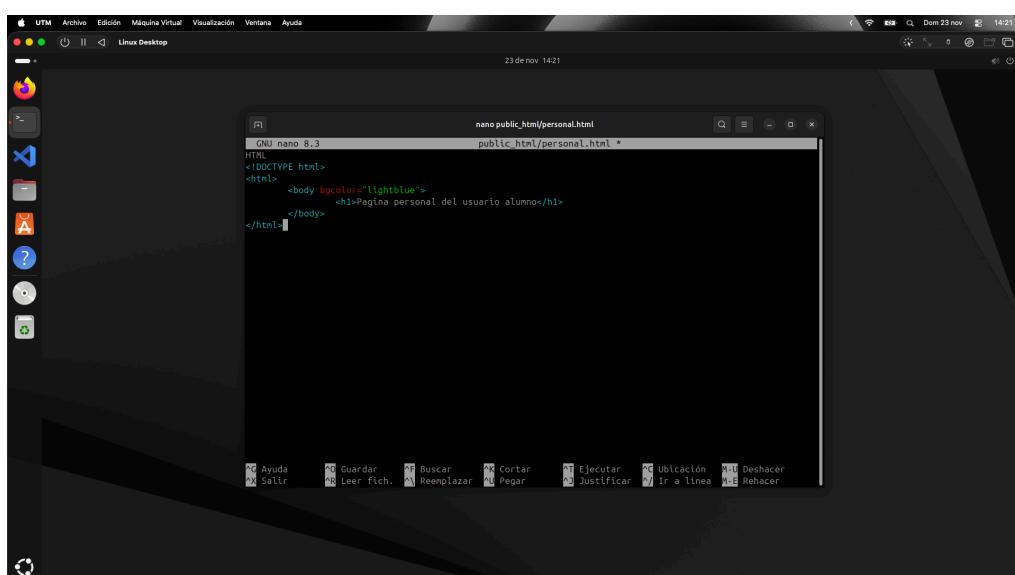
```
mkdir public_html
```

Crear el html

```
nano public_html/personal.html
```



Contenido de `personal.html` (sencillo pero con color como pide):



PASO 1.4: DAR PERMISOS

1. **Apache necesita permiso de ejecución (x) en tu carpeta de usuario para poder atravesarla y llegar a public_html:**

```
sudo chmod 755 /home/mario
```

2. **Asegura los permisos de la carpeta pública.** La carpeta donde guardas la web también debe ser accesible:

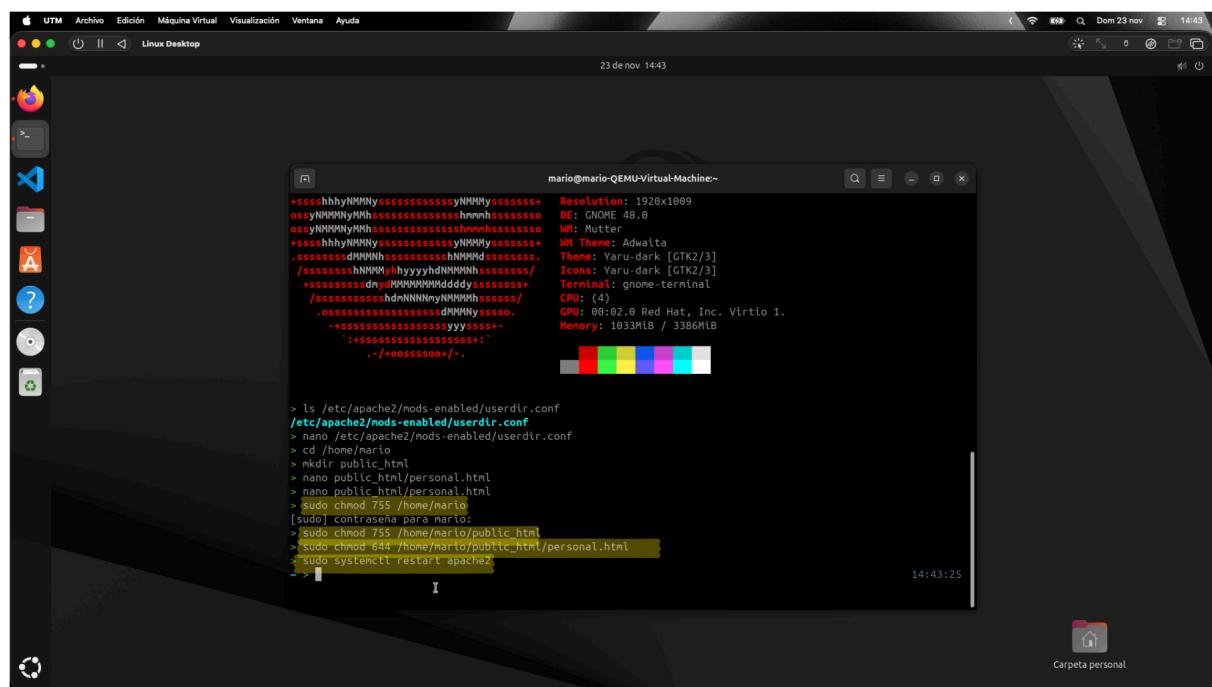
```
sudo chmod 755 /home/mario/public_html
```

3. **Asegura los permisos del archivo.** El archivo HTML debe poder ser leído:

```
sudo chmod 644 /home/mario/public_html/personal.html
```

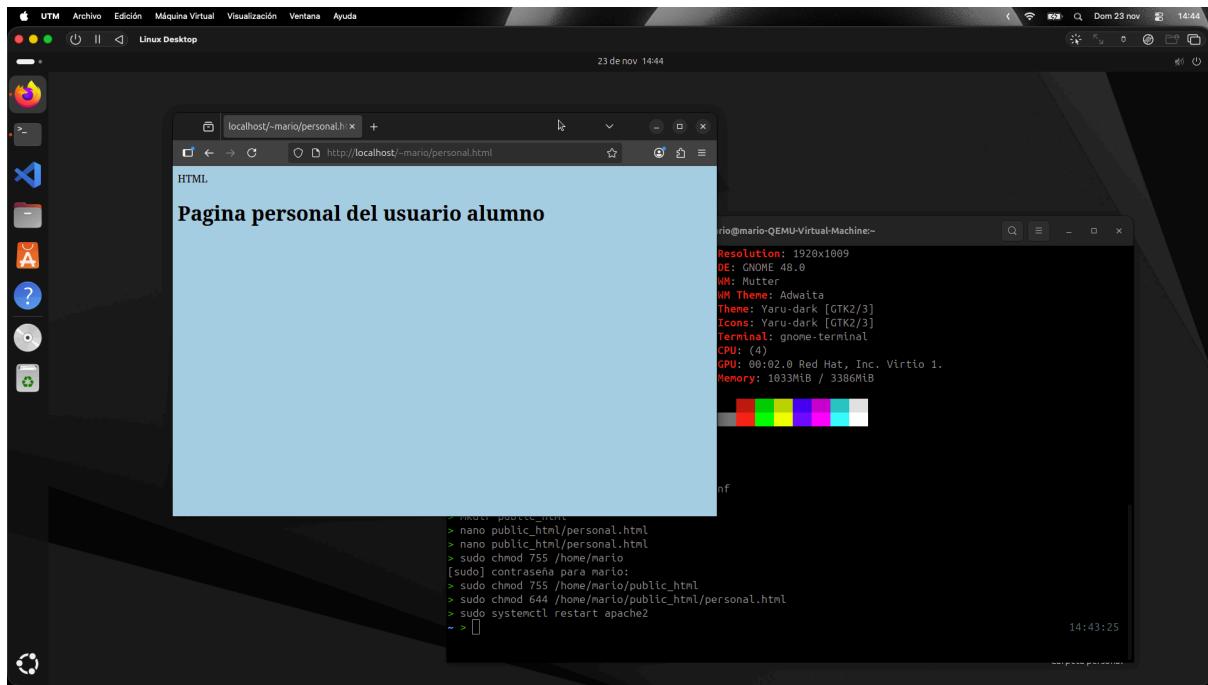
4. **Reinicia Apache (por si acaso)**

```
sudo systemctl restart apache2
```



PASO 1.5: COMPROBACIÓN

Abre el navegador en tu máquina virtual y entra en: <http://localhost/~mario/personal.html>



2. ZONA PRIVADA CON CONTRASEÑA (.htaccess)

Aquí vamos a proteger una carpeta para que pida usuario y contraseña.

PASO 2.1: CREAR DIRECTORIOS Y FICHEROS

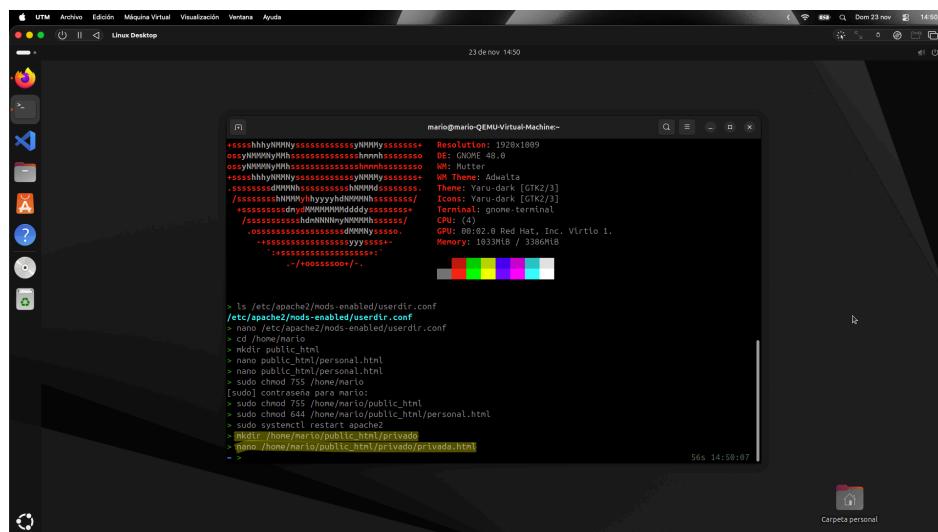
El enunciado pide que la carpeta privada cuelgue del directorio principal.

Crear carpeta

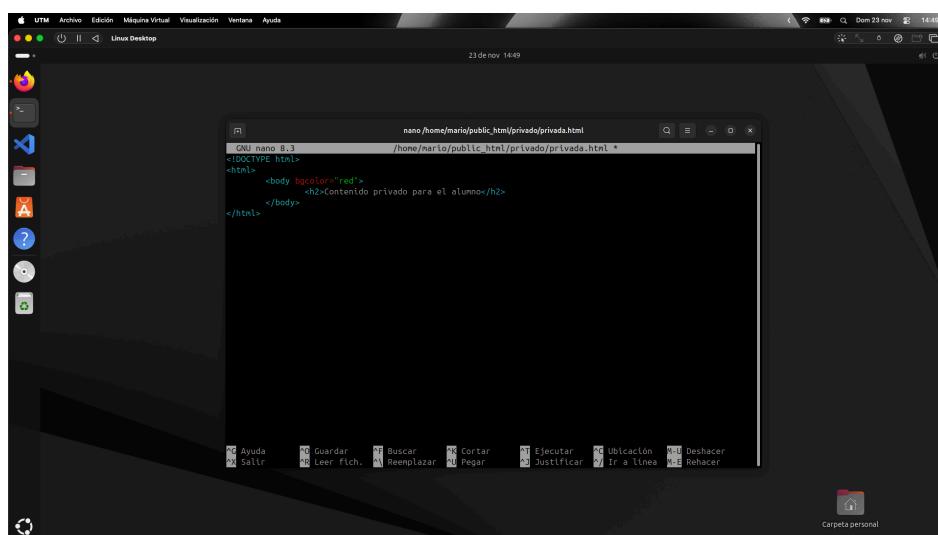
```
mkdir /home/mario/public_html/privado
```

Crear el html privado

```
nano /home/mario/public_html/privado/privada.html
```



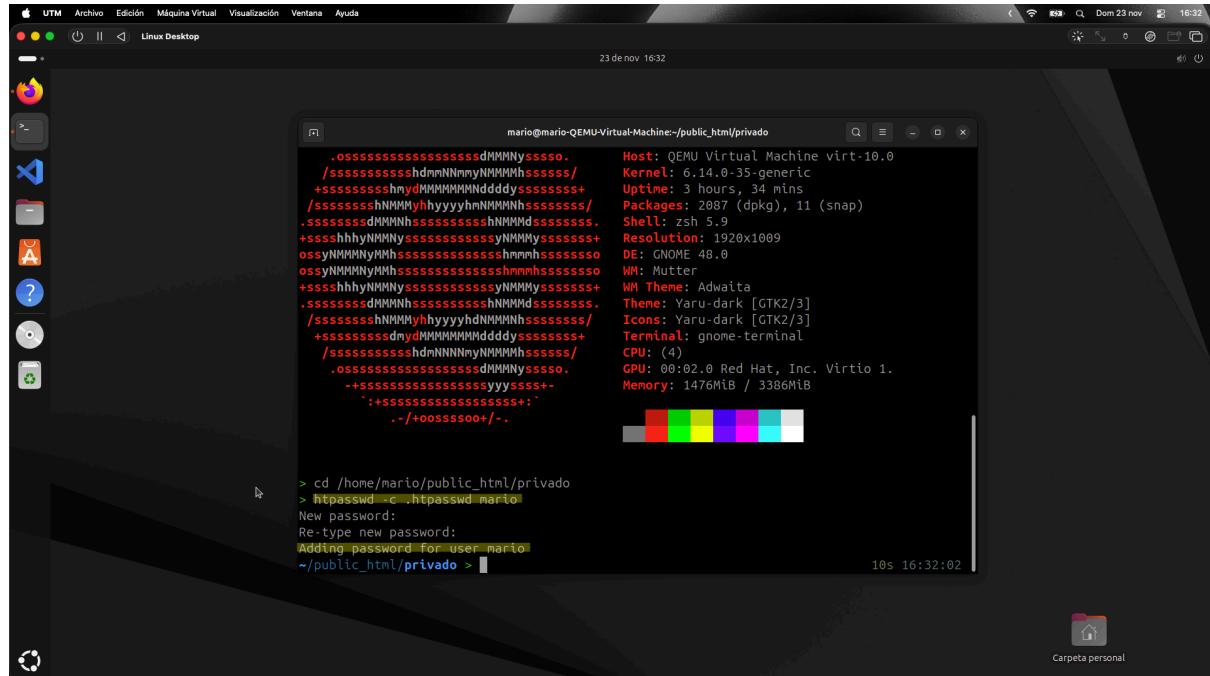
Contenido de `privada.html`:



PASO 2.2: CREAR EL USUARIO Y LA CONTRASEÑA

Usamos htpasswd. El PDF dice que guardemos el fichero de claves en el mismo directorio.

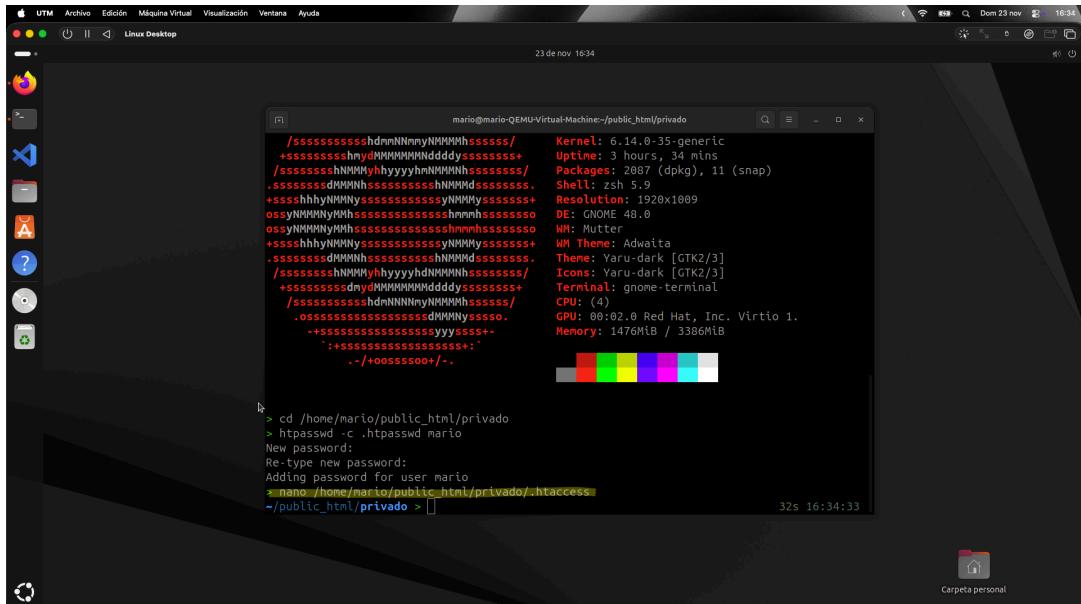
```
cd /home/mario/public_html/privado/  
htpasswd -c .htpasswd mario  
(te pedirá que escribas la contraseña dos veces. Ej: 1234)
```



PASO 2.3: CONFIGURAR EL FICHERO .HTACCESS

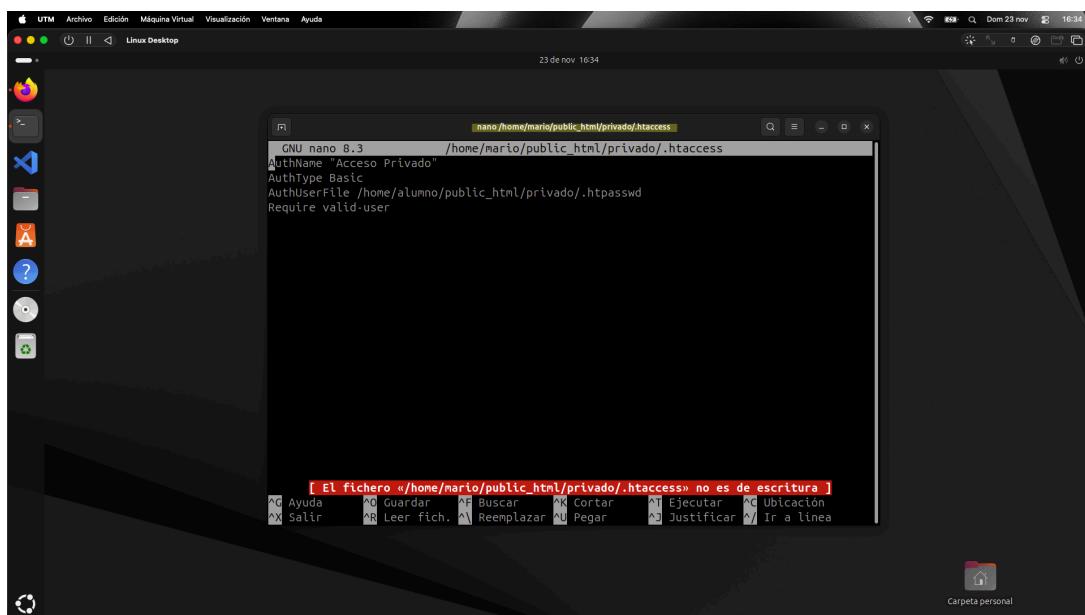
Dentro de la carpeta privado, creamos el fichero que controla el acceso:

```
nano /home/mario/public_html/privado/.htaccess
```



Contenido del .htaccess (Copia esto exacto):

```
AuthName "Acceso Privado"
AuthType Basic
AuthUserFile /home/alumno/public_html/privado/.htpasswd
Require valid-user
```



Configuración de errores personalizados (punto extra del examen)

ErrorDocument 404 "La página buscada no existe"

Truco del experto (vital para que funcione):

Por defecto, Apache ignora los ficheros .htaccess en las carpetas de usuario. Tienes que editar la configuración general para permitirlos.

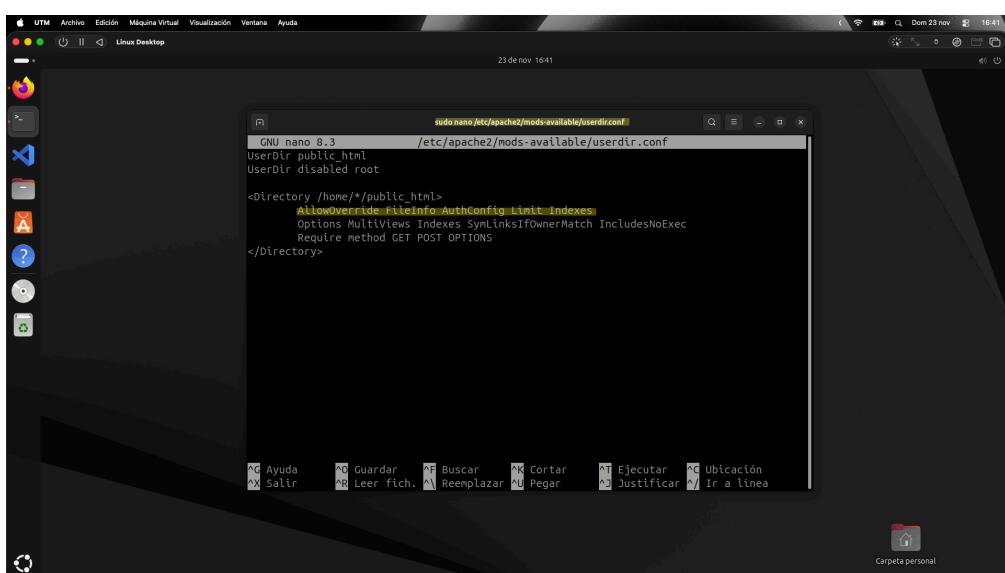
1. Edita: sudo nano /etc/apache2/mods-enabled/userdir.conf

2. Dentro del bloque <Directory /home/*/*public_html>, **cambia**

AllowOverride FileInfo AuthConfig Limit Indexes **a:**

AllowOverride All

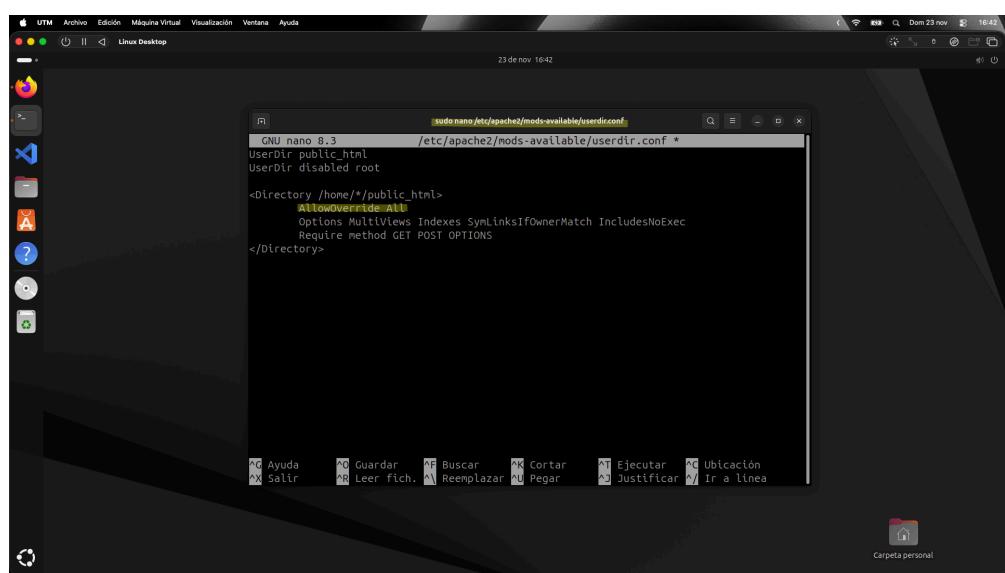
3. Reinicia Apache: sudo systemctl restart apache2



A screenshot of a Linux desktop environment showing a terminal window titled "sudo nano /etc/apache2/mods-available/userdir.conf". The window displays the Apache configuration code. The "AllowOverride" line is highlighted with a red box. The terminal window has a dark theme and includes standard nano keybindings at the bottom.

```
GNU nano 8.3 /etc/apache2/mods-available/userdir.conf
UserDir public_html
UserDir disabled root

<Directory /home/*/*public_html>
    AllowOverride FileInfo AuthConfig Limit Indexes
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    Require method GET POST OPTIONS
</Directory>
```



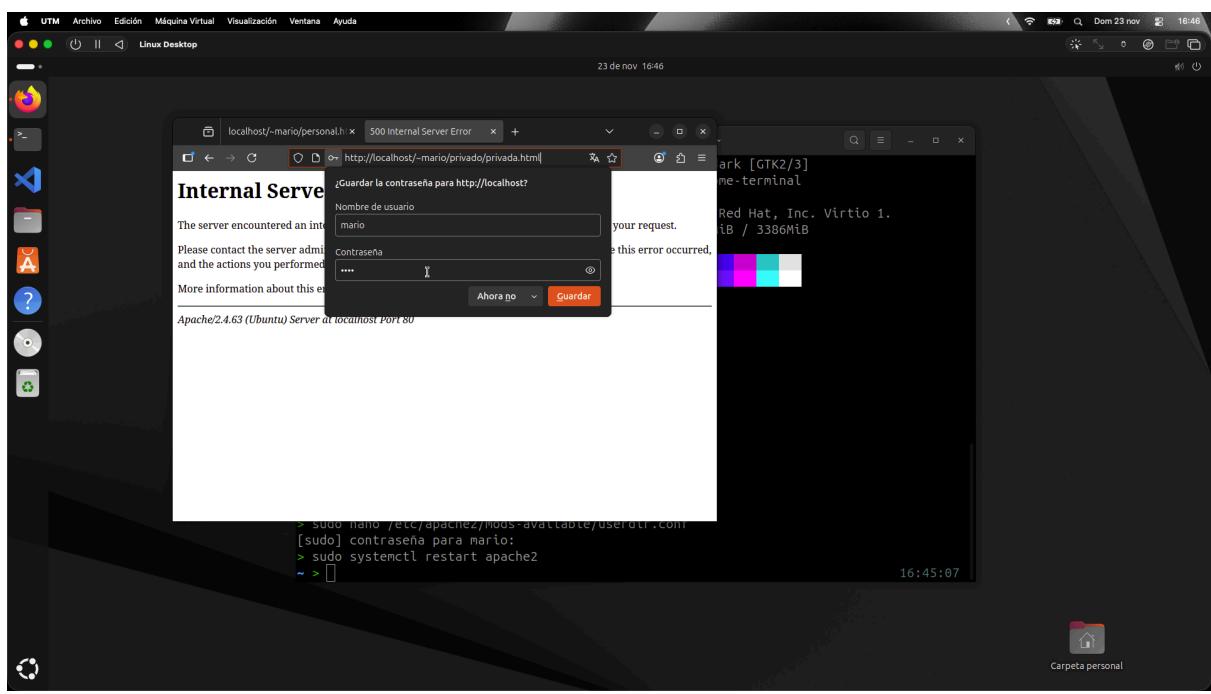
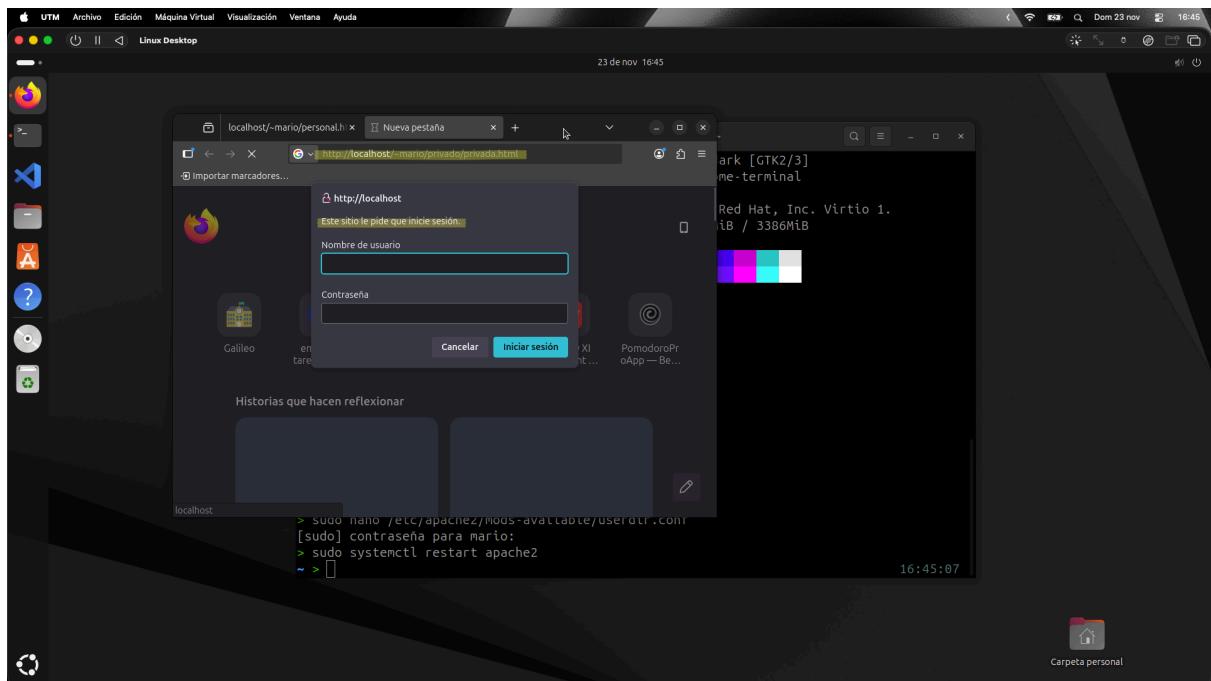
A screenshot of a Linux desktop environment showing the same terminal window after the configuration has been modified. The "AllowOverride" line now contains "All" instead of "FileInfo". The terminal window has a dark theme and includes standard nano keybindings at the bottom.

```
GNU nano 8.3 /etc/apache2/mods-available/userdir.conf *
UserDir public_html
UserDir disabled root

<Directory /home/*/*public_html>
    AllowOverride All
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    Require method GET POST OPTIONS
</Directory>
```

PASO 2.4: COMPROBACIÓN

Entra en <http://localhost/~mario/privado/privada.html>. Te debe saltar la ventanita pidiendo usuario y contraseña.



PUNTO 3: CONFIGURACIÓN DEL SITIO WEB CON DIRECTORIOS VIRTUALES (Wiki)

PASO 3.1: Preparar los ficheros de la Wiki

Primero vamos a crear el contenido. El enunciado dice que la wiki debe estar en tu carpeta personal (`/home/mario/wiki`), no en la carpeta del servidor.

1. Abre una terminal (asegúrate de no ser root, o sal de root con `exit` si lo eras).

Ve a tu carpeta personal:

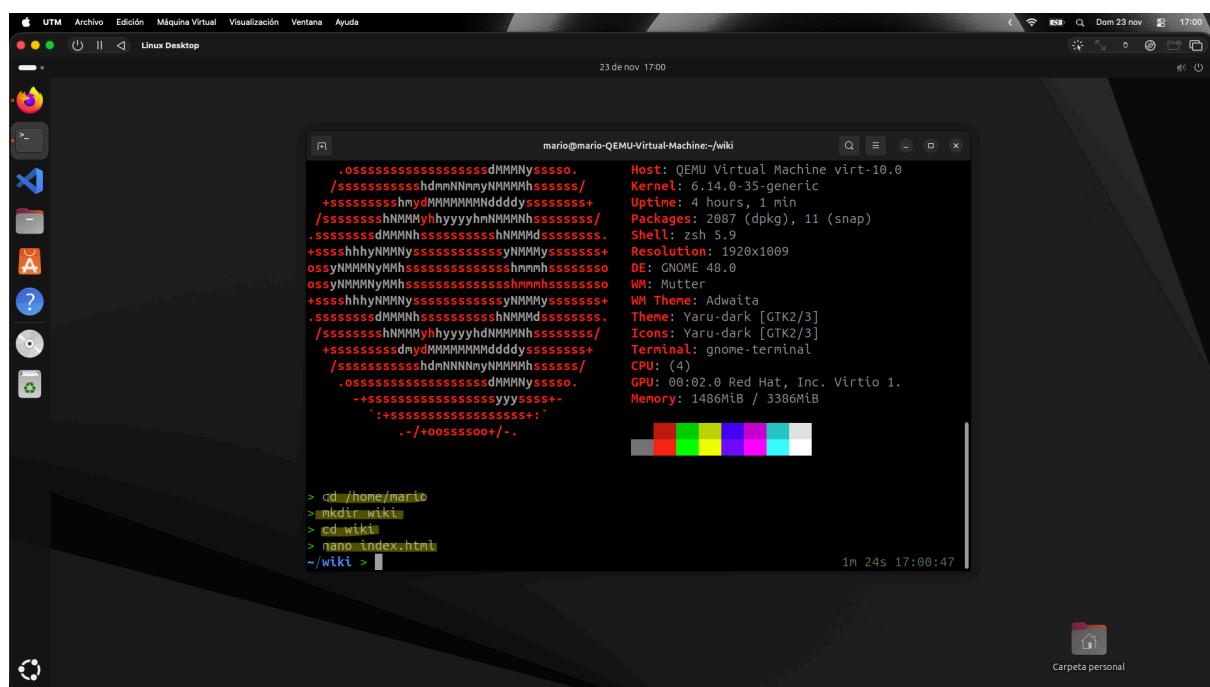
```
cd /home/mario
```

Crea el directorio para la wiki:

```
mkdir wiki
```

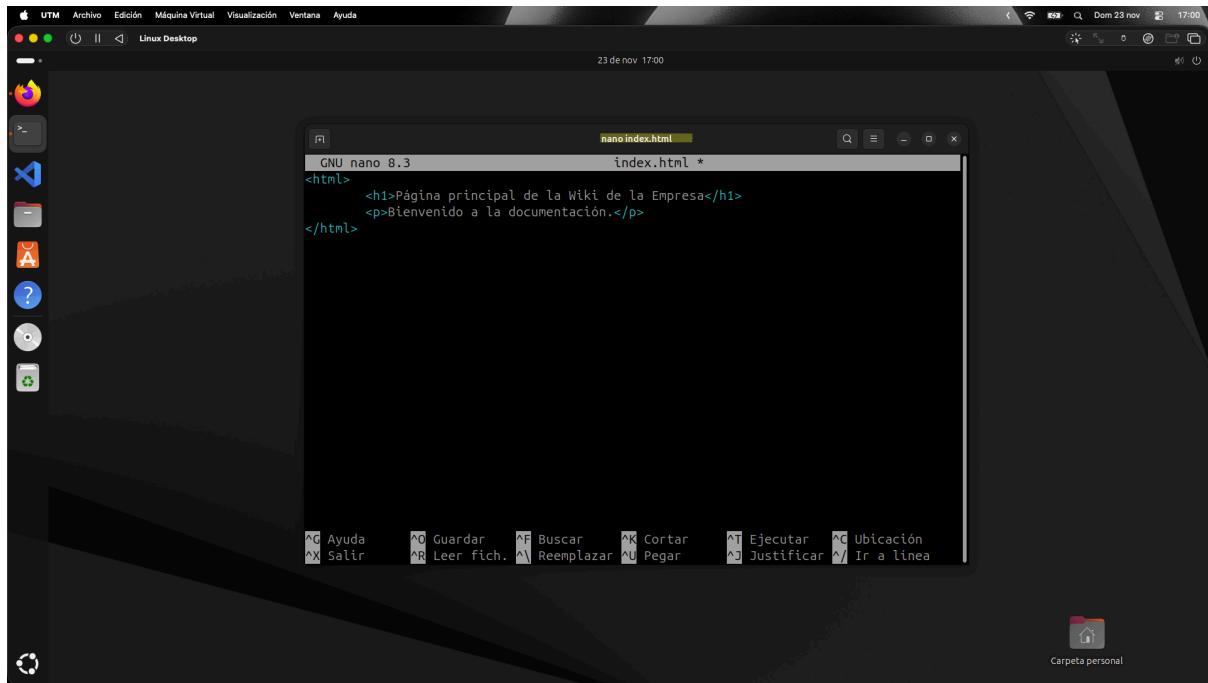
Entra en el directorio:

```
cd wiki
```



Crea el fichero de la página principal:

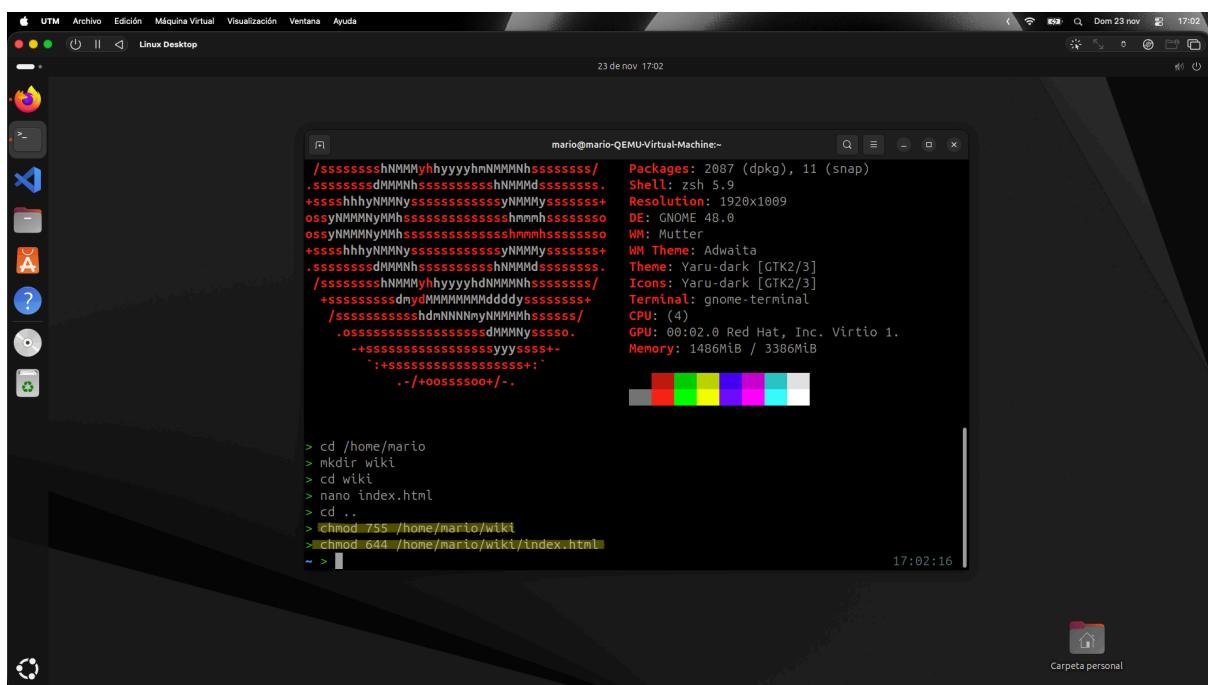
nano index.html



IMPORTANTE (para evitar el error 403): Dale permisos a esta nueva carpeta para que Apache pueda entrar.

chmod 755 /home/mario/wiki

chmod 644 /home/mario/wiki/index.html



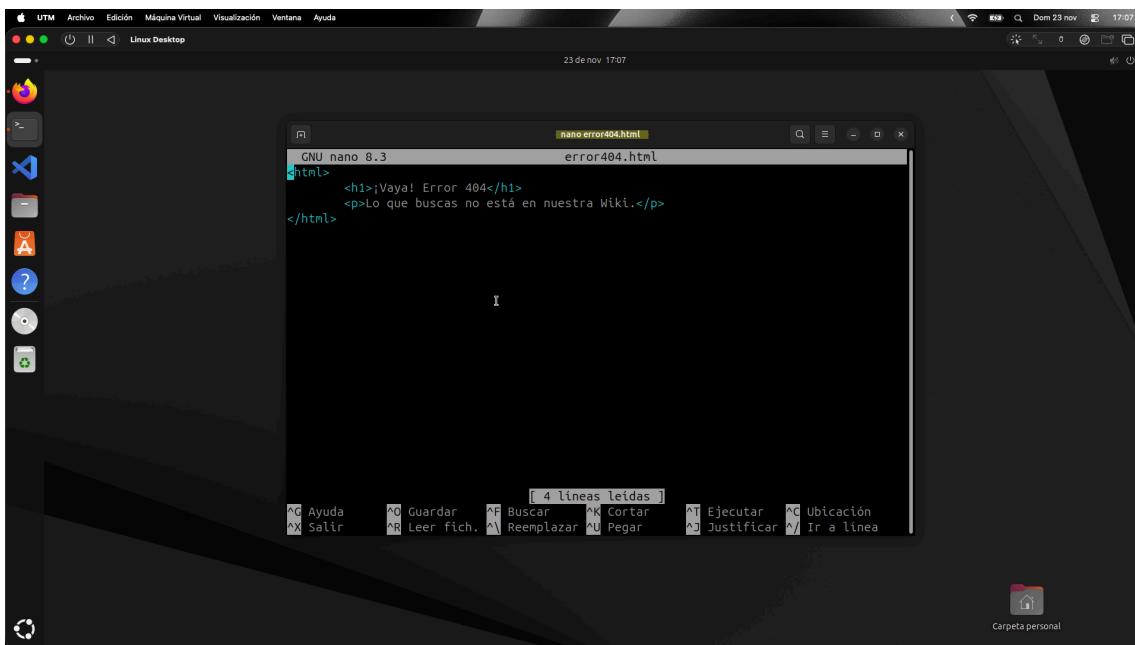
Paso 3.2: CREAR LA PÁGINA DE ERROR PERSONALIZADA

El enunciado pide "*Configura los errores típicos... más amigables*". Vamos a crear una página para cuando alguien busque algo que no existe (Error 404).

Sigues dentro de la carpeta wiki ([/home/mario/wiki](#)).

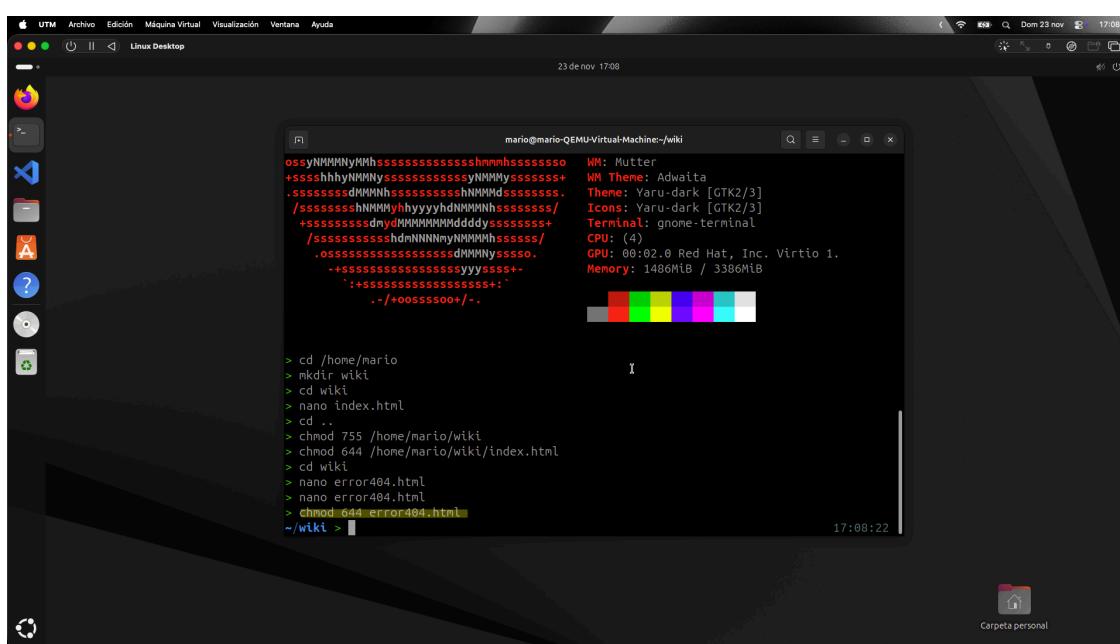
Crea el archivo de error:

```
sudo nano error404.html
```



Dale permisos tambié:

```
chmod 644 error404.html
```



PASO 3.3: CREAR EL ENLACE SIMBÓLICO

Ahora vamos a engañar a Apache. Apache busca en `/var/www/html`, pero nosotros tenemos la web en `/home/alumno/wiki`. Crearemos un "acceso directo" (enlace simbólico).

1. Ahora necesitas permisos de administrador. Usa `sudo` delante de los comandos.

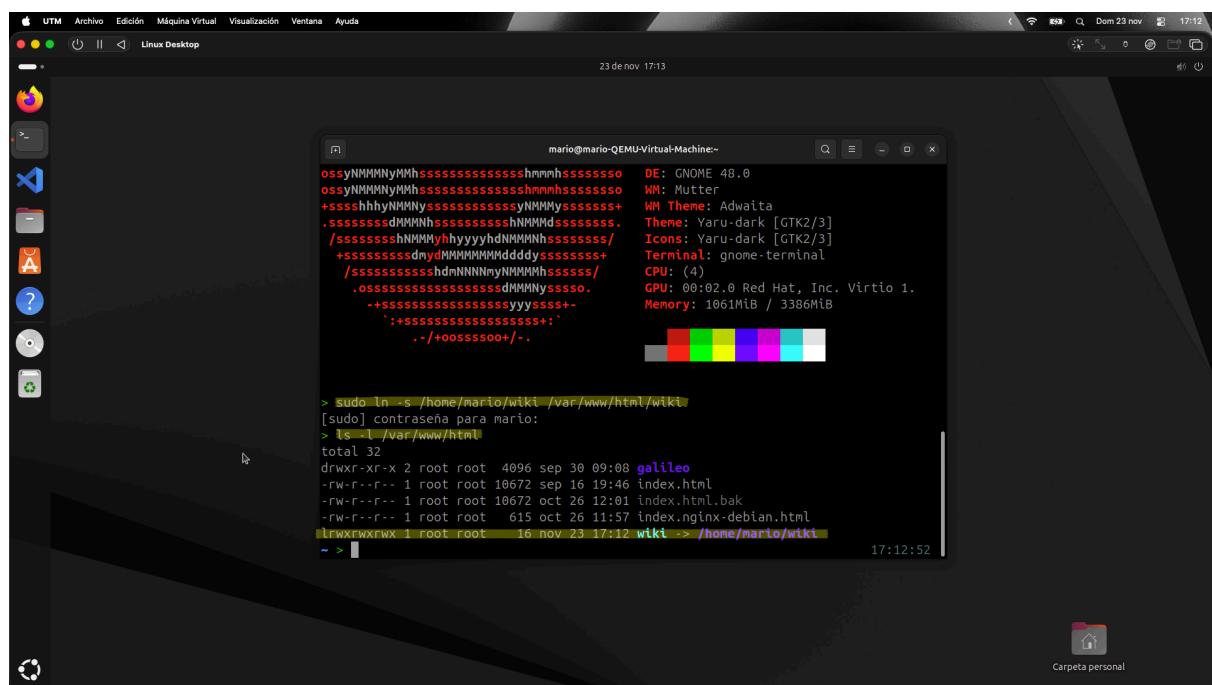
Crea el enlace:

```
sudo ln -s /home/mario/wiki /var/www/html/wiki
```

(esto significa: "Cuando alguien vaya a `/var/www/html/wiki`, mándalo realmente a `/home/mario/wiki`").

2. Ejecuta este comando para demostrar que el enlace está creado:

```
ls -l /var/www/html
```



PASO 3.4: CREAR LA CONFIGURACIÓN DEL SITIO (wiki.conf)

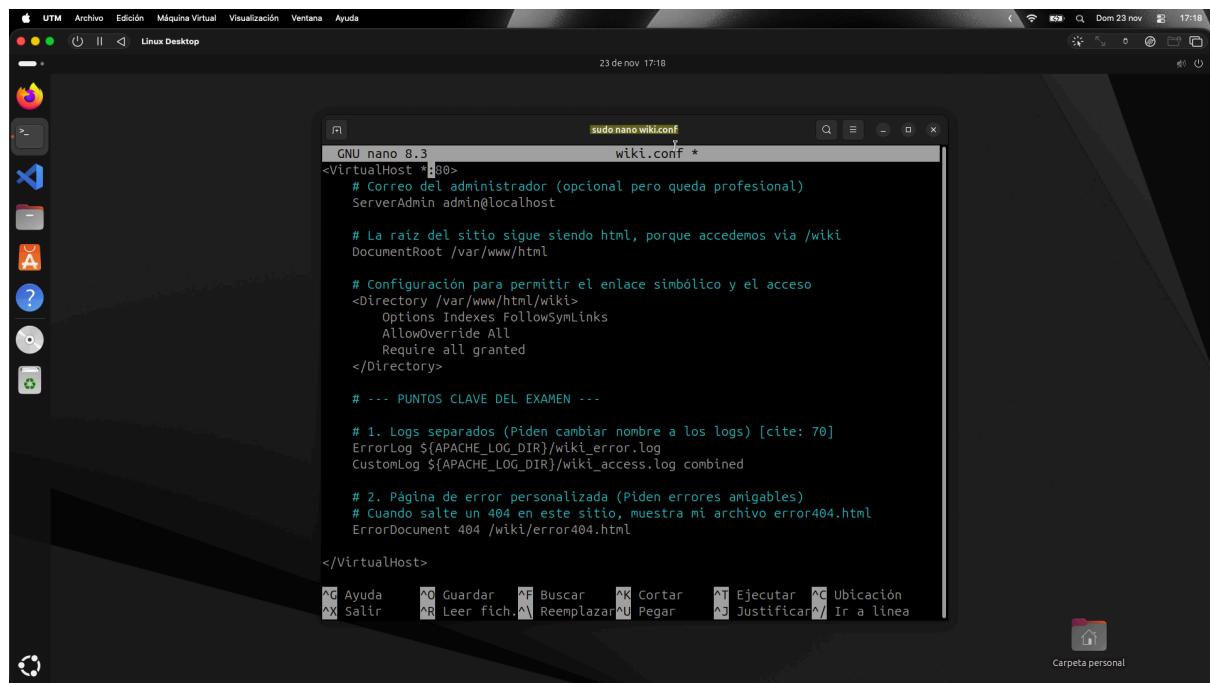
El enunciado pide crear el "sitio wiki" y configurar logs propios .

Ve a la carpeta de sitios de Apache:

```
cd /etc/apache2/sites-available/
```

Crea el fichero de configuración (con logs propios, directivas y página de error):

```
sudo nano wiki.conf
```



```
GNU nano 8.3          wiki.conf *
<VirtualHost *:80>
    # Correo del administrador (opcional pero queda profesional)
    ServerAdmin admin@localhost

    # La raíz del sitio sigue siendo html, porque accedemos via /wiki
    DocumentRoot /var/www/html/wiki

    # Configuración para permitir el enlace simbólico y el acceso
    <Directory /var/www/html/wiki>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    # --- PUNTOS CLAVE DEL EXAMEN ---

    # 1. Logs separados (Piden cambiar nombre a los logs) [cite: 70]
    ErrorLog ${APACHE_LOG_DIR}/wiki_error.log
    CustomLog ${APACHE_LOG_DIR}/wiki_access.log combined

    # 2. Página de error personalizada (Piden errores amigables)
    # Cuando salte un 404 en este sitio, muestra mi archivo error404.html
    ErrorDocument 404 /wiki/error404.html

</VirtualHost>
```

PASO 3.5: ACTIVAR EL SITIO Y REINICIAR

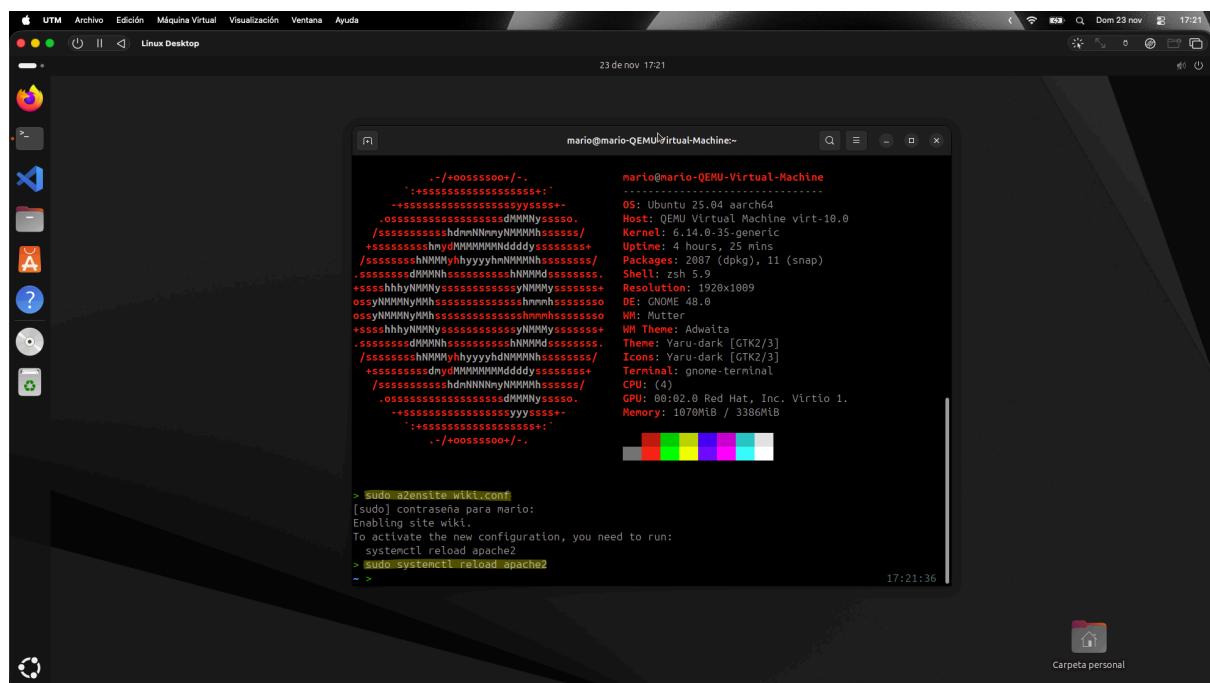
Para que esto funcione bien y no se mezcle con la configuración por defecto, lo más "limpio" para el examen es desactivar el sitio por defecto y activar el tuyo, o simplemente activar el tuyo y que Apache gestione la mezcla. Vamos a hacerlo activando el tuyo.

Activa tu nuevo sitio:

```
sudo a2ensite wiki.conf
```

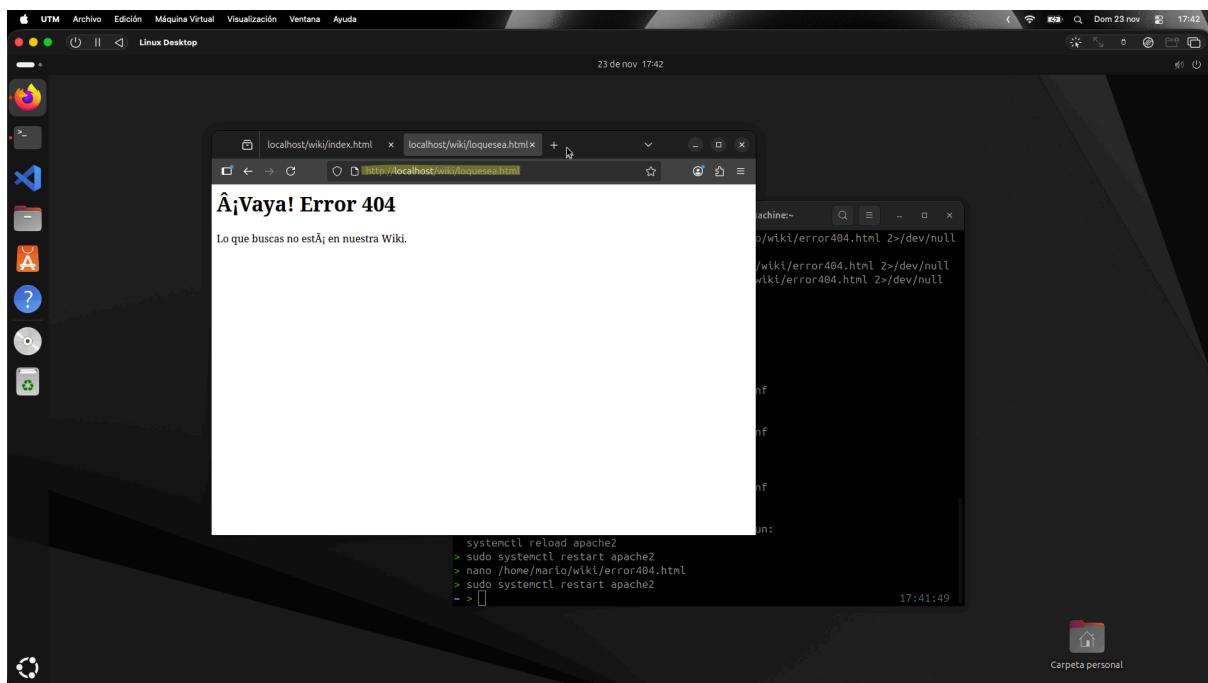
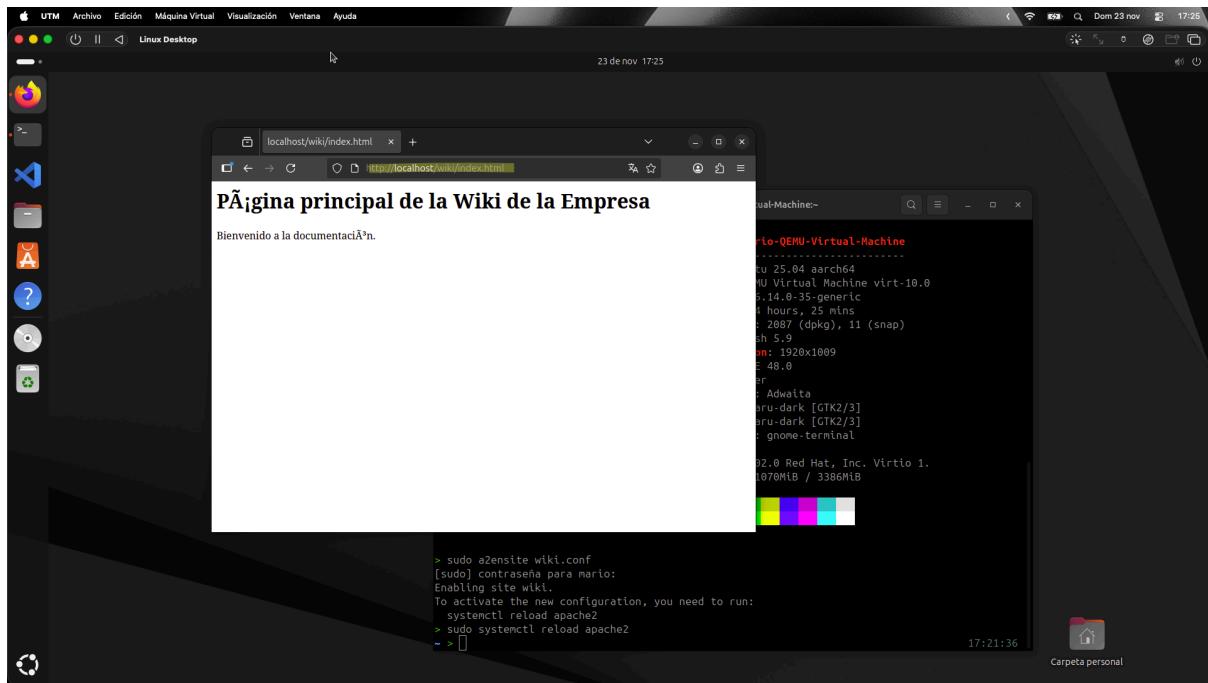
Recarga Apache para aplicar cambios:

```
sudo systemctl reload apache2
```



Paso 3.6: Comprobación Final y Capturas de Logs

1. Abre el navegador de la máquina virtual.
2. Entra en: <http://localhost/wiki/index.html> (deberías ver tu "Página principal de la Wiki")
3. **Prueba el error:** Intenta entrar en una página inventada, por ejemplo: <http://localhost/wiki/loquesea.html> (debería salir tu página de "¡Vaya! Error 404")



Verificar los Logs (Punto crítico del examen):

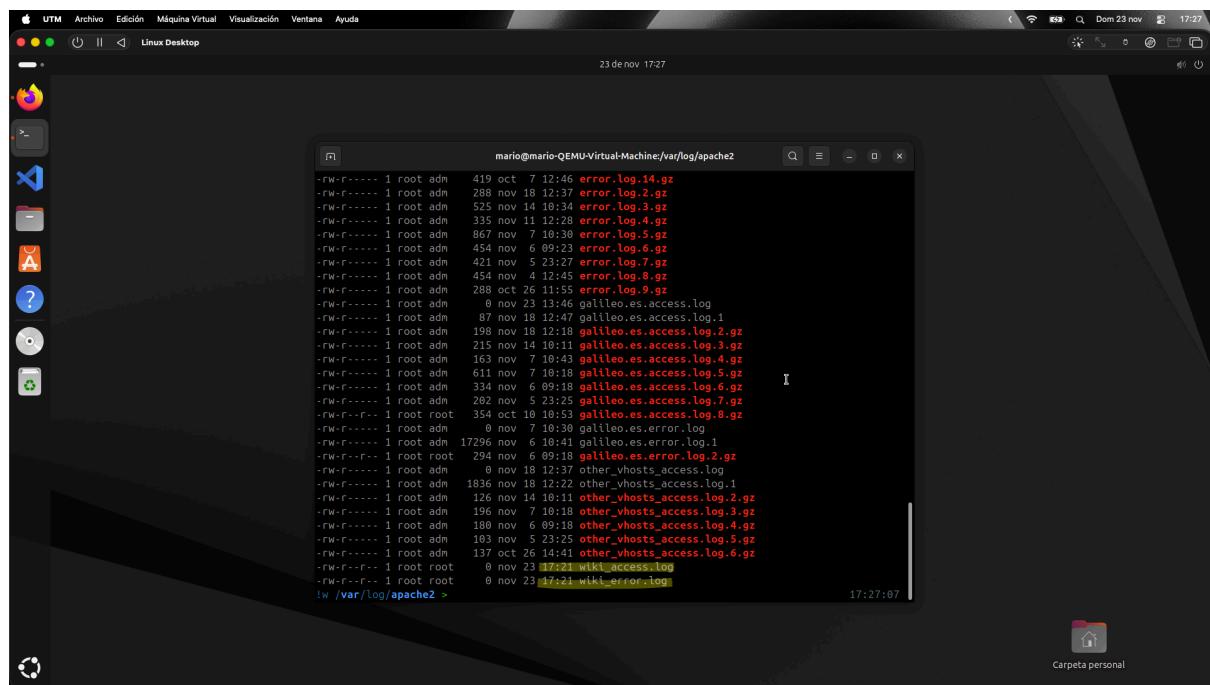
Ahora que has entrado, Apache ha tenido que escribir en los ficheros de log nuevos que definimos. Vamos a verlos.

Ve a la carpeta de logs:

```
cd /var/log/apache2/
```

Lista los archivos:

```
ls -l
```



The screenshot shows a terminal window titled "mario@mario-QEMU-Virtual-Machine:/var/log/apache2" running on a dark-themed desktop environment. The terminal displays a long list of files in the /var/log/apache2 directory, showing their permissions, owner, group, size, date modified, and name. The files are mostly compressed log files like "error.log.14.gz" through "error.log.9.gz" and "error.log". There are also several "access.log" files, some named by date (e.g., "access.log.1", "access.log.2.gz") and others by host ("gallileo.es.access.log", "other_vhosts.access.log"). The terminal window has a scroll bar and is positioned in the center of the screen. The desktop background is a dark abstract image, and the taskbar at the top includes icons for UTM, Archivo, Edición, Máquina Virtual, Visualización, Ventana, and Ayuda. A small "Linux Desktop" icon is visible in the bottom right corner of the terminal window.

```
-rw-r----- 1 root adm 419 oct 7 12:46 error.log.14.gz  
-rw-r----- 1 root adm 288 nov 18 12:37 error.log.2.gz  
-rw-r----- 1 root adm 525 nov 14 10:34 error.log.3.gz  
-rw-r----- 1 root adm 335 nov 11 12:28 error.log.4.gz  
-rw-r----- 1 root adm 867 nov 7 10:30 error.log.5.gz  
-rw-r----- 1 root adm 454 nov 6 09:23 error.log.6.gz  
-rw-r----- 1 root adm 421 nov 5 23:27 error.log.7.gz  
-rw-r----- 1 root adm 454 nov 4 12:45 error.log.8.gz  
-rw-r----- 1 root adm 288 oct 26 11:55 error.log.9.gz  
-rw-r----- 1 root adm 0 nov 23 13:45 gallileo.es.access.log  
-rw-r----- 1 root adm 87 nov 18 12:47 gallileo.es.access.log.1  
-rw-r----- 1 root adm 198 nov 18 12:18 gallileo.es.access.log.2.gz  
-rw-r----- 1 root adm 215 nov 14 10:11 gallileo.es.access.log.3.gz  
-rw-r----- 1 root adm 163 nov 7 10:43 gallileo.es.access.log.4.gz  
-rw-r----- 1 root adm 611 nov 7 10:18 gallileo.es.access.log.5.gz  
-rw-r----- 1 root adm 334 nov 6 09:18 gallileo.es.access.log.6.gz  
-rw-r----- 1 root adm 202 nov 5 23:25 gallileo.es.access.log.7.gz  
-rw-r----- 1 root adm 354 oct 10 10:53 gallileo.es.access.log.8.gz  
-rw-r----- 1 root adm 0 nov 7 10:30 gallileo.es.error.log  
-rw-r----- 1 root adm 17296 nov 6 10:41 gallileo.es.error.log.1  
-rw-r----- 1 root root 294 nov 6 09:18 gallileo.es.error.log.2.gz  
-rw-r----- 1 root adm 0 nov 18 12:37 other_vhosts.access.log  
-rw-r----- 1 root adm 1836 nov 18 12:22 other_vhosts.access.log.1  
-rw-r----- 1 root adm 126 nov 14 10:11 other_vhosts.access.log.2.gz  
-rw-r----- 1 root adm 196 nov 7 10:18 other_vhosts.access.log.3.gz  
-rw-r----- 1 root adm 189 nov 6 09:18 other_vhosts.access.log.4.gz  
-rw-r----- 1 root adm 103 nov 5 23:25 other_vhosts.access.log.5.gz  
-rw-r----- 1 root adm 137 oct 26 14:41 other_vhosts.access.log.6.gz  
-rw-r----- 1 root root 0 nov 23 17:21 wiki1.access.log  
-rw-r----- 1 root root 0 nov 23 17:21 wiki_error.log  
tw /var/log/apache2 >
```