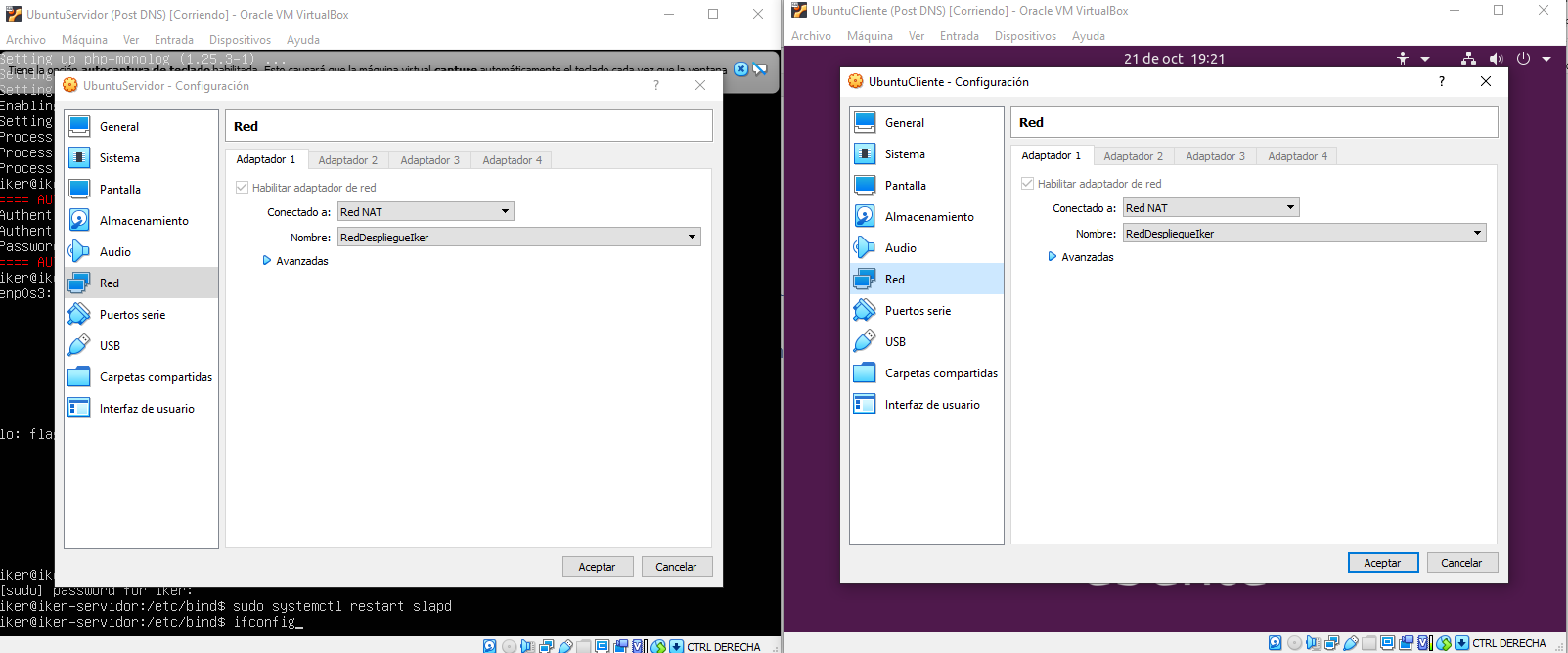
# A) Tenemos este escenario inicial:

## a. Trabajaremos con un Ubuntu y un cliente Ubuntu (reutiliza los que ya tenías y adjunta captura).

## b. Las dos VM están conectadas entre sí mediante una red común (adjunta captura).



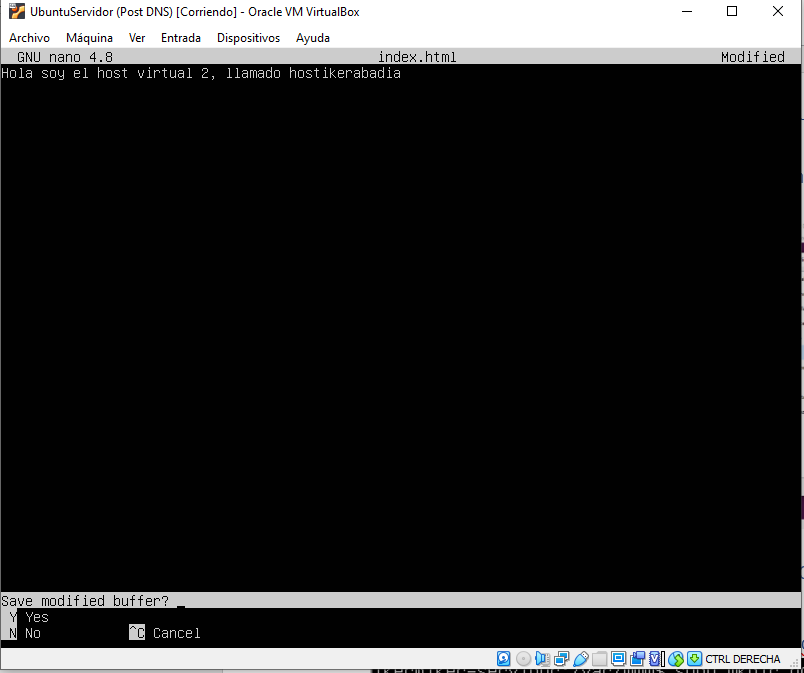
# B) Vamos a crear un host virtual basado en nombre llamado host<ikerabadia>

## a. Crea un directorio bajo /var/www llamado “hostikerabadia”.

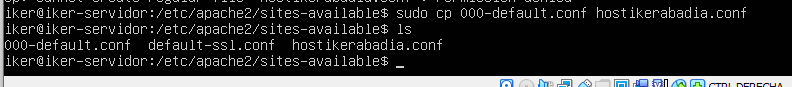


## b. Crea index.html dentro del nuevo directorio y escribe en él: “Hola, soy el host virtual 2, llamado “hostikerabadia”

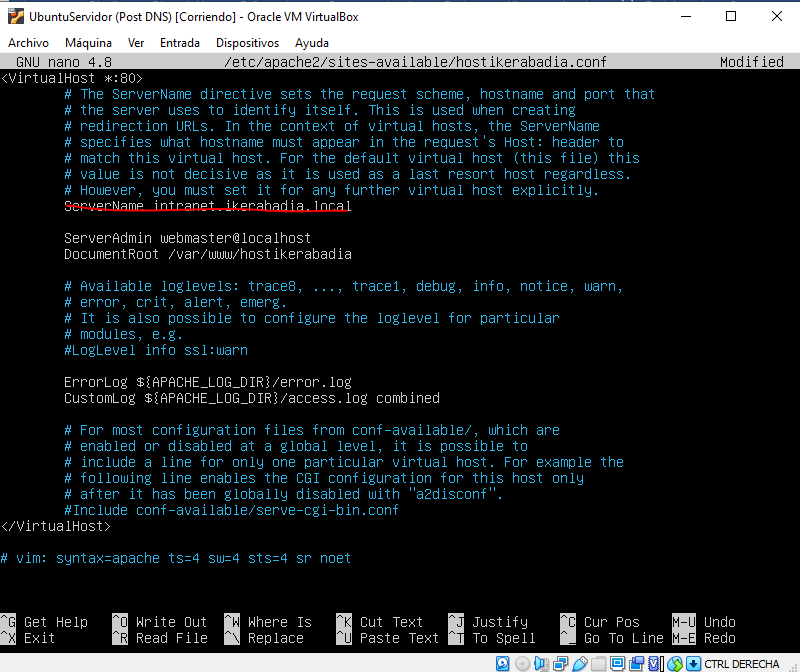




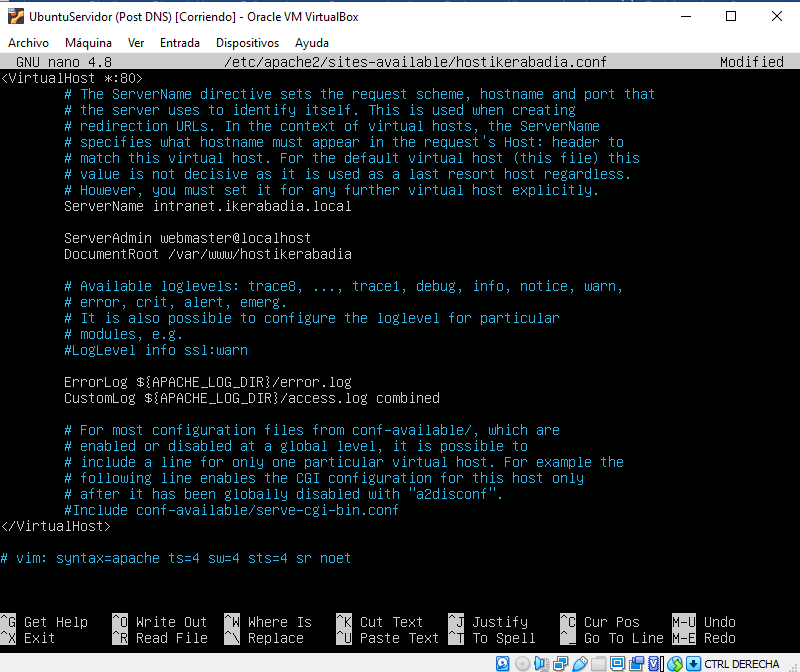
## c. Duplica el archivo por defecto /etc/apache2/sites-available/000-default.conf y llámalo hostikerabadia.conf



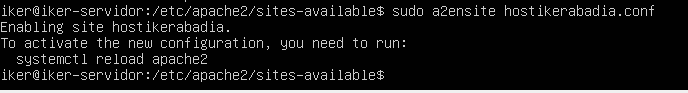
## d. Edita el nuevo archivo y actualiza DocumentRoot al directorio que aplica.



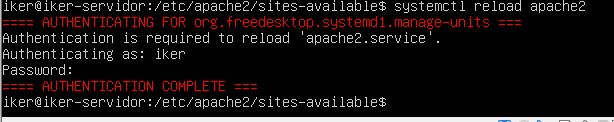
## e. En ServerName indica intranet.ikerabadia.local



## f. Ejecuta “a2ensite hostikerabadia.conf”

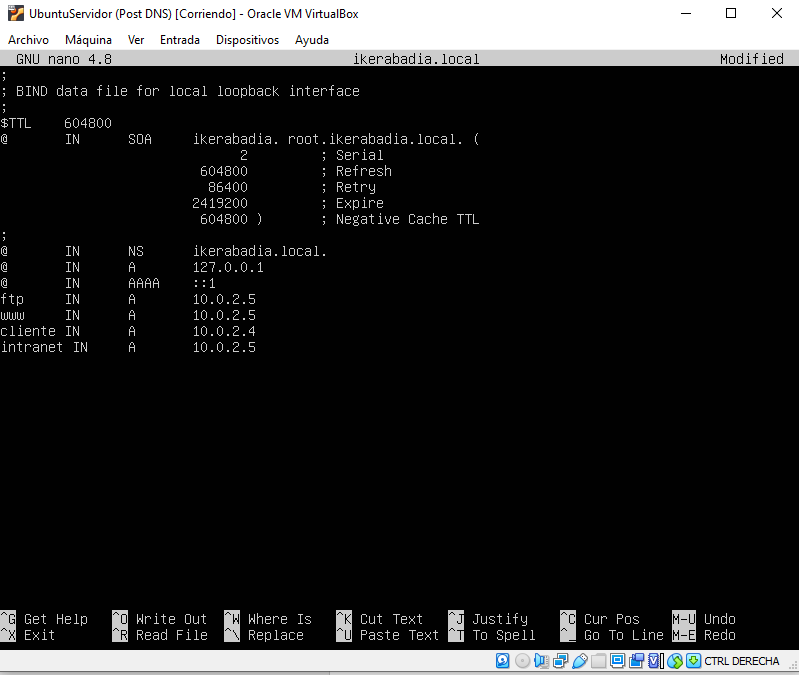


## g. Reinicia el servicio.

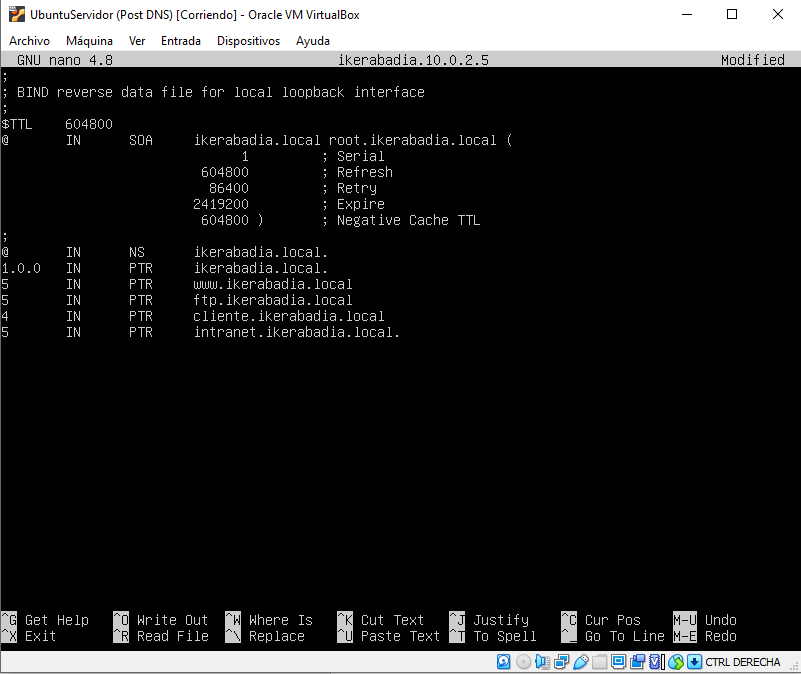


## h. Crea una entrada DNS para el subdominio intranet.ikerabadia.local en el servicio Bind9

para ello lo primero que tengo que hacer es ir a /etc/bind



Esto (solo la última linea) va en ikerabadia.local

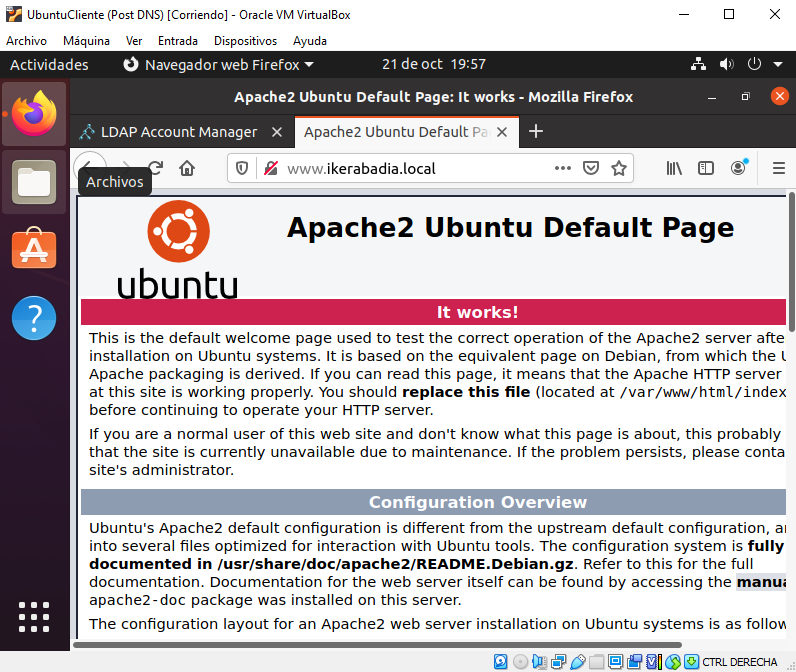


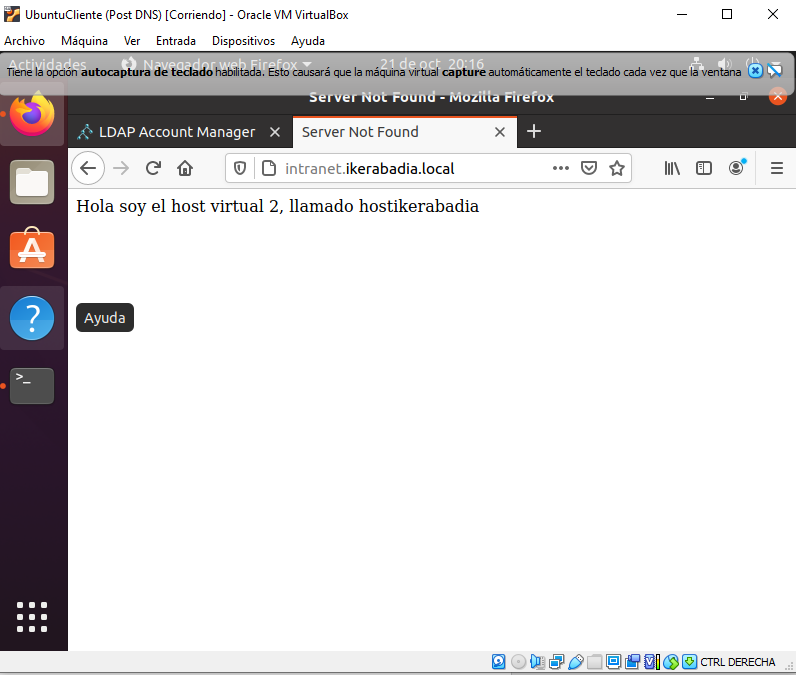
Y esto en (solo la última linea) ikerabadia.10.0.2.5

## i. Verifica desde el cliente, con el navegador web y por consola, que www.ikerabadia.local e intranet.ikerabadia.local están configurados y renderizan dos páginas web distintas. Es decir, tienes ya dos host virtuales basados en nombre.

**PARA HACER ESTO TENGO QUE REINICIAR EL SERVICIO systemctl restart bind9**



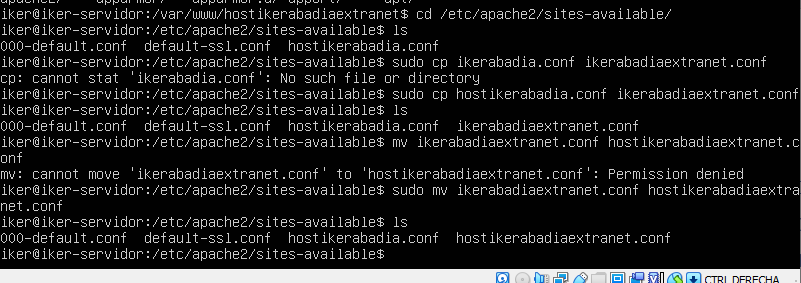




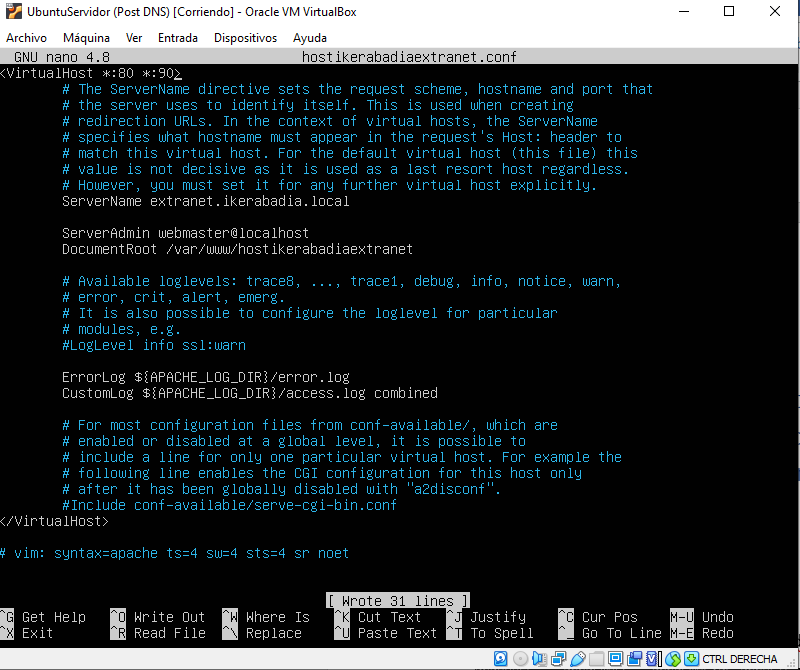
# C) Crea un nuevo host virtual escuchando en el puerto 90 y en el puerto 80 para el subdominio denominado extranet.ikerabadia .local, genera un index.html con contenido diferenciado. Realiza las acciones pertinentes y demuestra que funciona todo verificando que renderizas la página web en cada puerto respectivo.

En la carpeta /var/www creo una carpeta hostikerabadiaextranet, después le creo un index.html dentro.

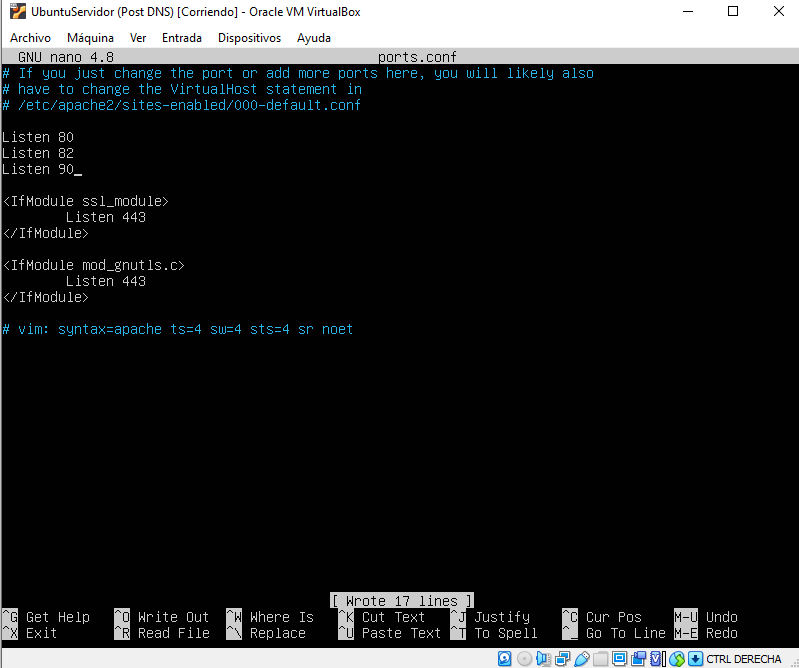
Voy al sites-available y copio hostikerabadia.conf para usarlo como plantilla



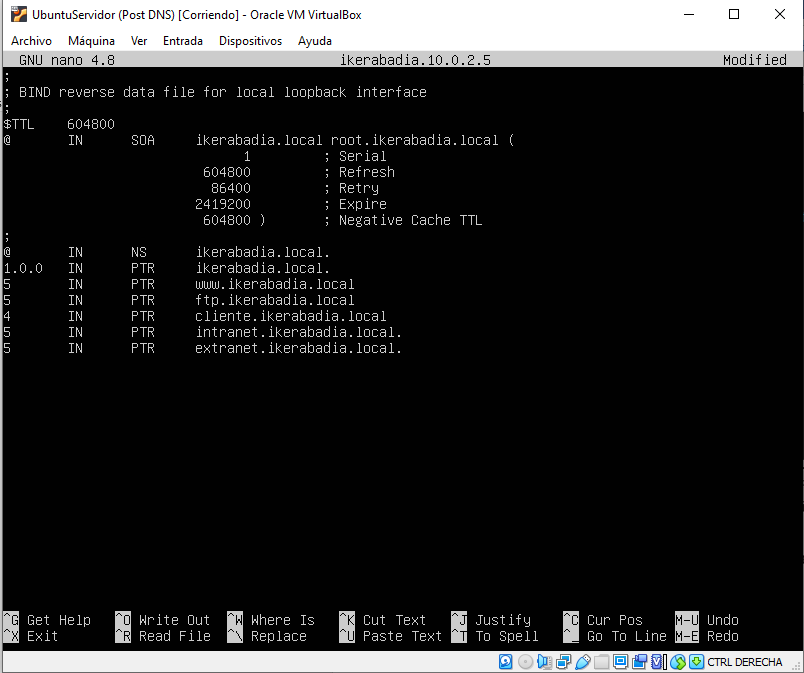
Lo edito y lo dejo de la siguiente manera:

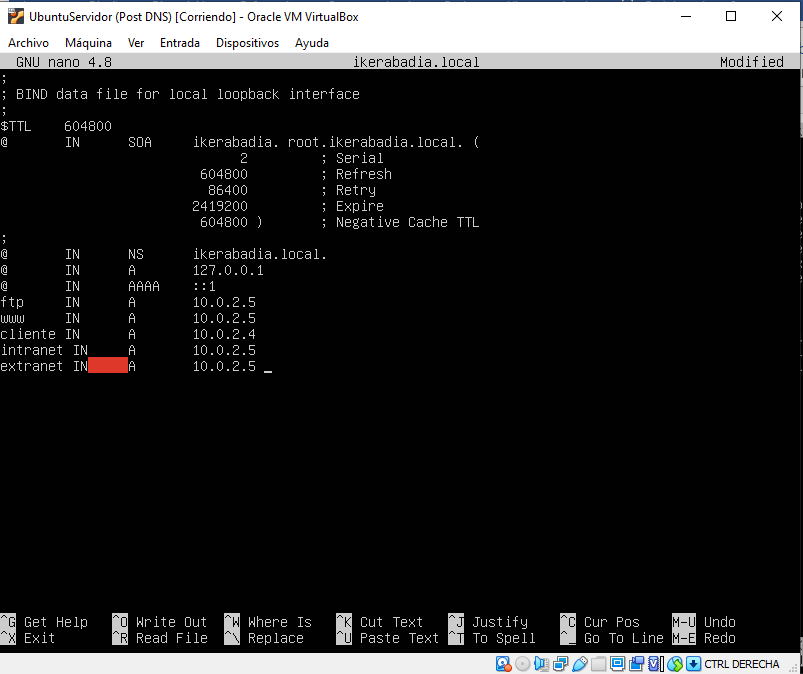


Cambio el ports.conf en /etc/apache2 y le añado la escucha en un nuevo puerto

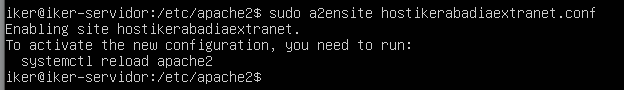


Y ahora hay que ir a /etc/bind y editar el archivo ikerabadia.10.0.2.5 (zona inversa) y el ikerabadia.local (zona directa)



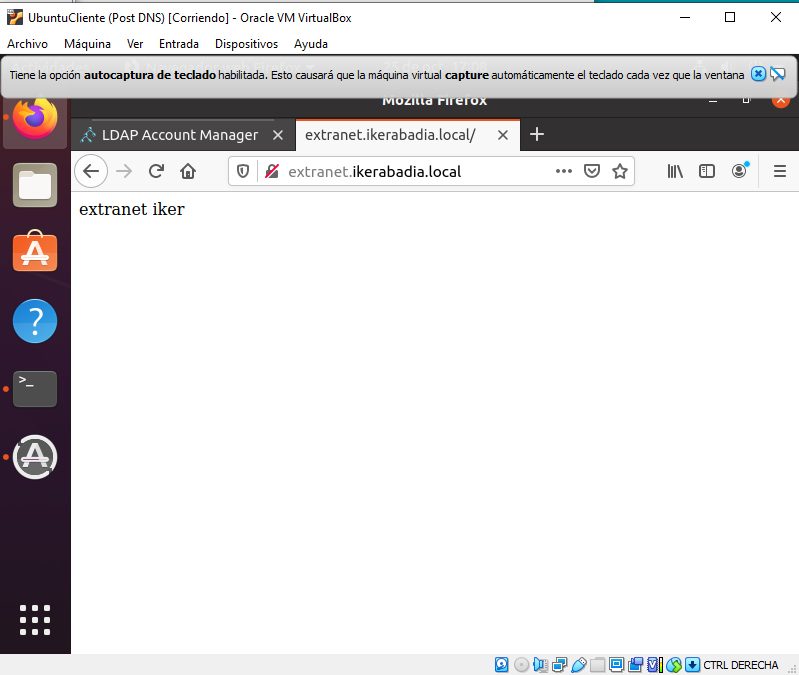


Ahora uso el comando a2ensite

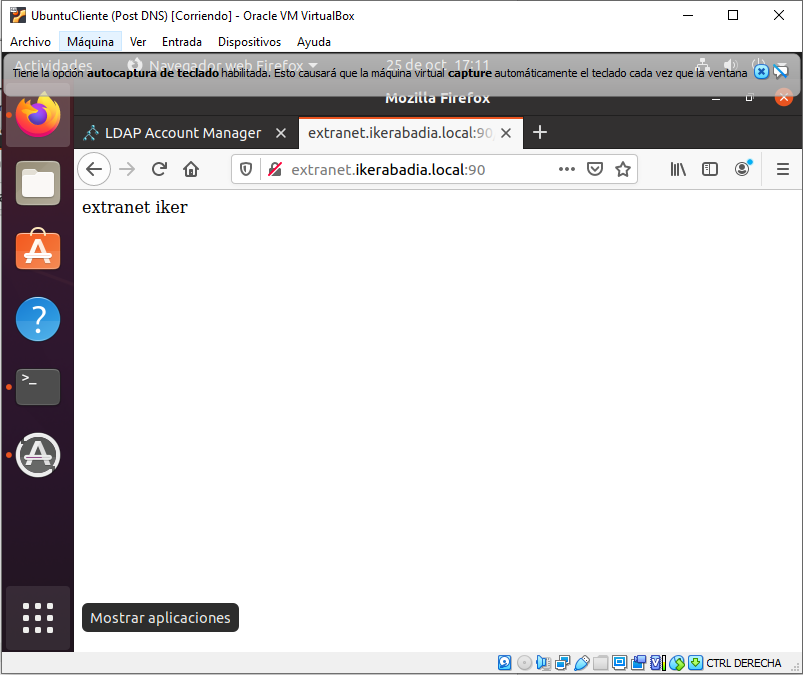


Y reinicio los servicios



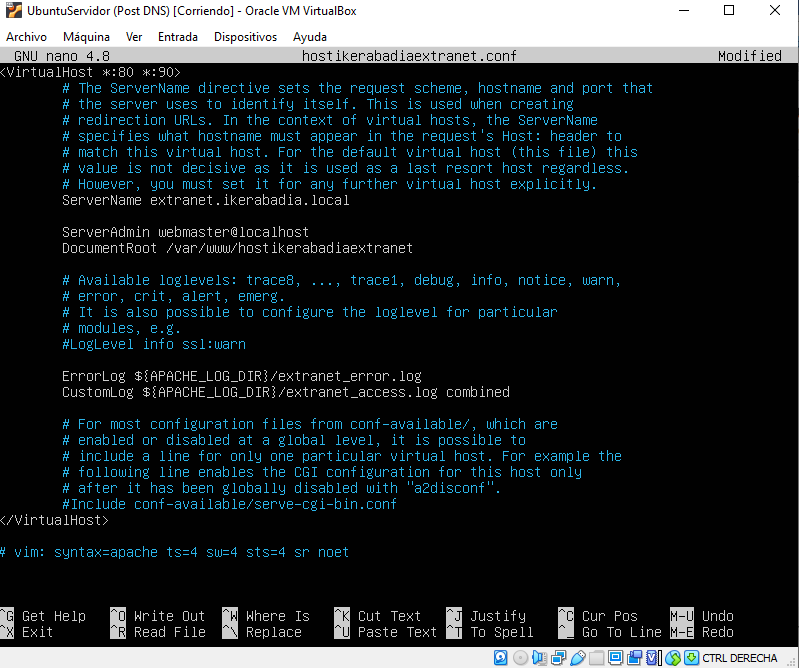


Aquí está el puerto 80



Y aquí el puerto 90

# D) Para el host virtual de la extranet, customiza el log de acceso y el log de errores llamándolos extranet\_error.log y extranet\_access.log respectivamente. Prueba que, en efecto, los logs se alimentan correctamente cuando se producen accesos y cuando se produce un error de ejecución.

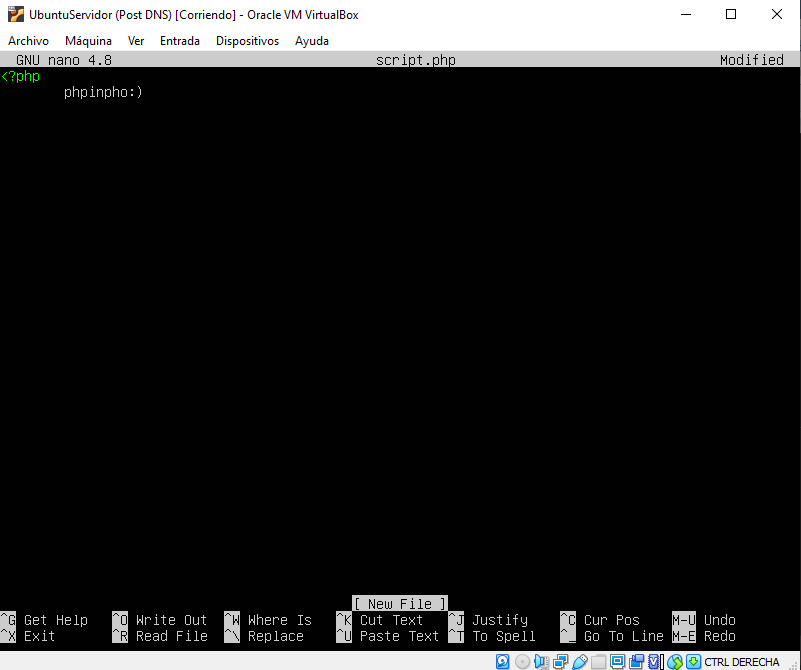


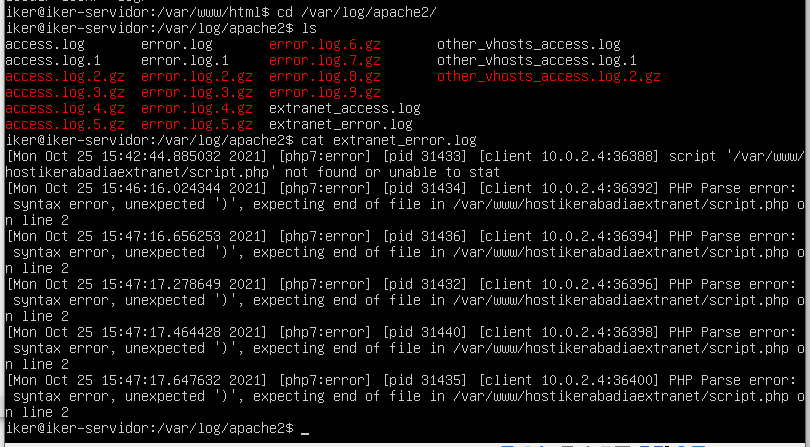
Edito el fichero de configuración del host que se encuentra en /etc/apache2/sites-available/(hostikerabadiaextranet.conf) y lo dejo como en la captura anterior.

Y reinicio el servicio

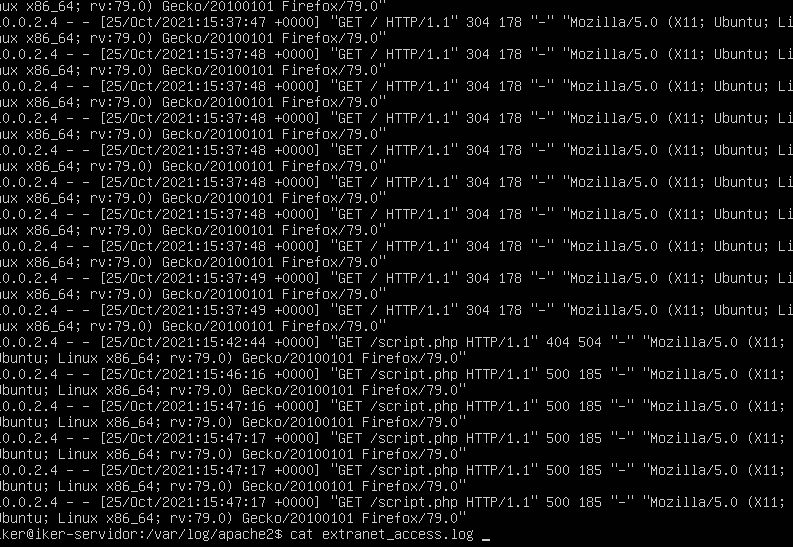


Ahora para comprobar que funcionan voy a /var/www/hostikerabadiaextanet y creo un script.php que de fallo.





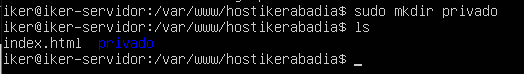
Aquí los errores



Y aquí los accesos

# E) Autenticación básica:

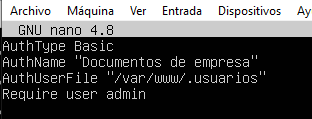
## a. Dentro del host virtual intranet crea un directorio llamado privado.



## b. Crea un archivo .htaccess dentro del directorio “privado” de este estilo:



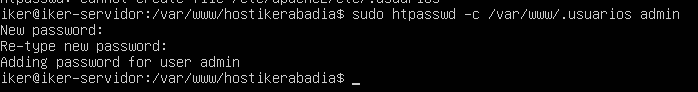
NOTA: cambiar /etc/apache2/etc/.usuarios por /var/www/.usuarios



## c. Para crear el archivo de usuario ejecuta:



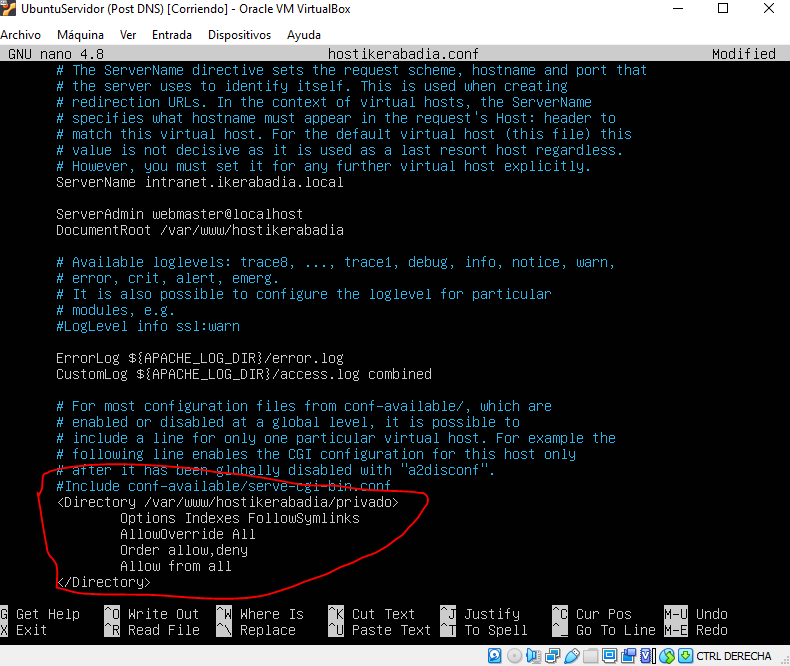
NOTA: cambiar /etc/apache2/etc/.usuarios por /var/www/.usuarios



## d. Configuraremos en el host virtual el acceso al directorio:

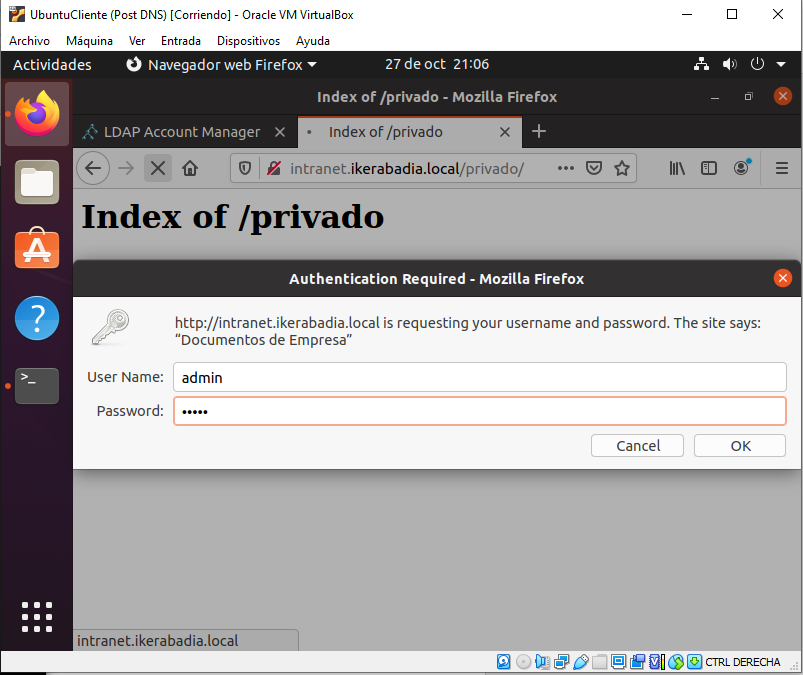


**NOTA : el directorio es “privado”, no “rrhh”**



## e. Reinicia el servicio y verifica que puedes acceder al directorio con el usuario indicado desde el navegador web del cliente.

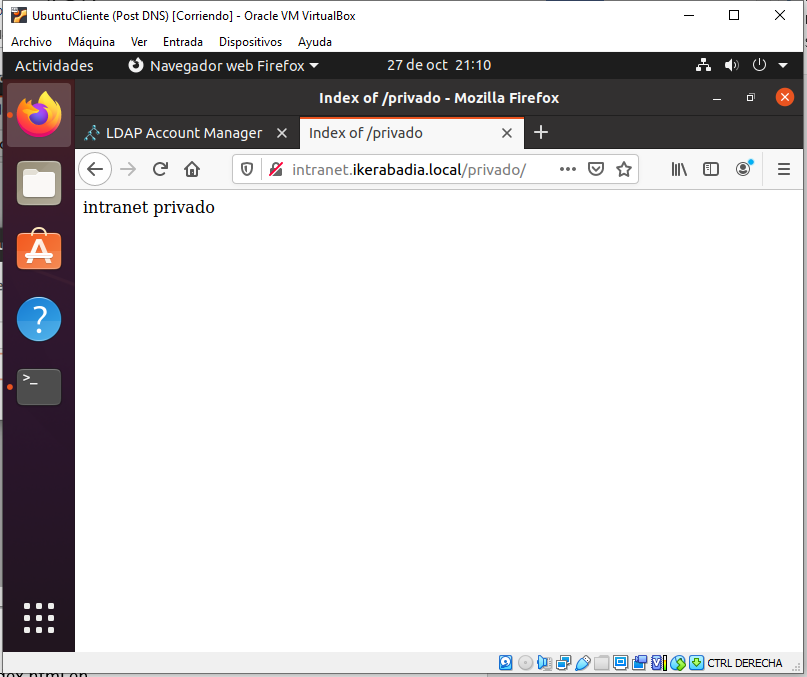




EN ESTE CASO USER->ADMIN PASSWORD->ADMIN

Para que se vea algo tras el logueo he creado un index.html en /var/www/hostikerabadia/privado



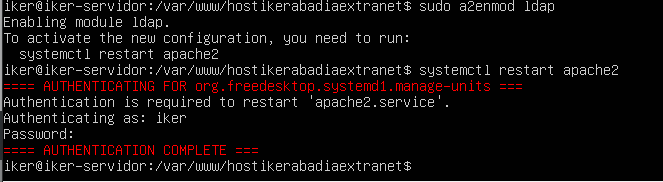


# F) Autenticación mediante LDAP:

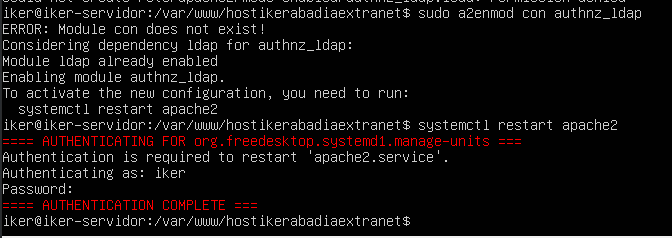
## a. Dentro del host de la extranet crea un directorio llamado ldap



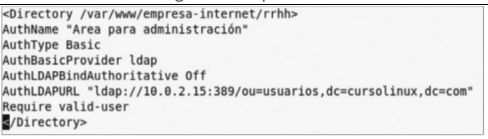
## b. Habilita el módulo ldap con: a2enmod ldap



## c. Habilita el módulo authnz\_ldap con: a2enmod con authnz\_ldap



## d. Configura en el host de la extranet un nuevo directorio similar a lo siguiente:



<Directory /var/www/hostikerabadiaextranet/ldap>

AuthName “Area para administracion”

AuthType Basic

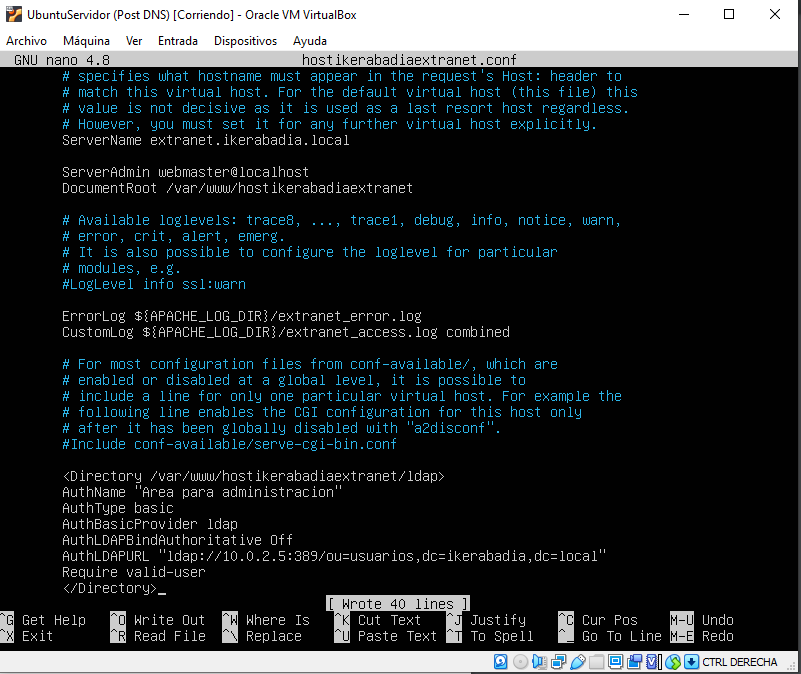
AuthBasicProvider ldap

AuthLDAPBindAuthoritative Off

AuthLDAPURL “ldap://10.0.2.5:389/ou=usuarios,dc=ikerabadia,dc=local”

Require valid-user

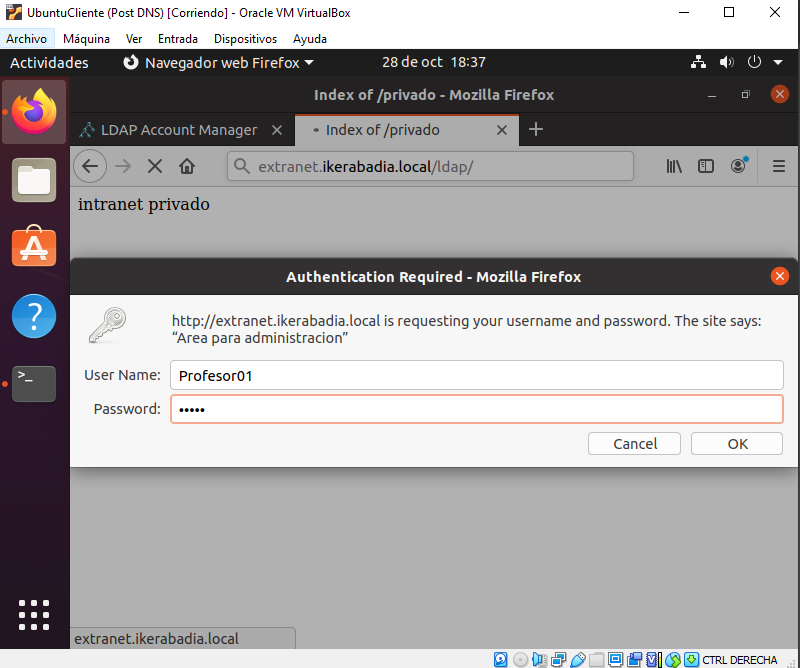
</Directory>



## e. Verifica que puedes entrar con cualquier usuario LDAP registrado en la unidad organizativa usuarios, vía autenticación básica en el navegador web del cliente.

**NOTA: ANTES DE NADA REINICIAR EL SERVICIO DE APACHE**



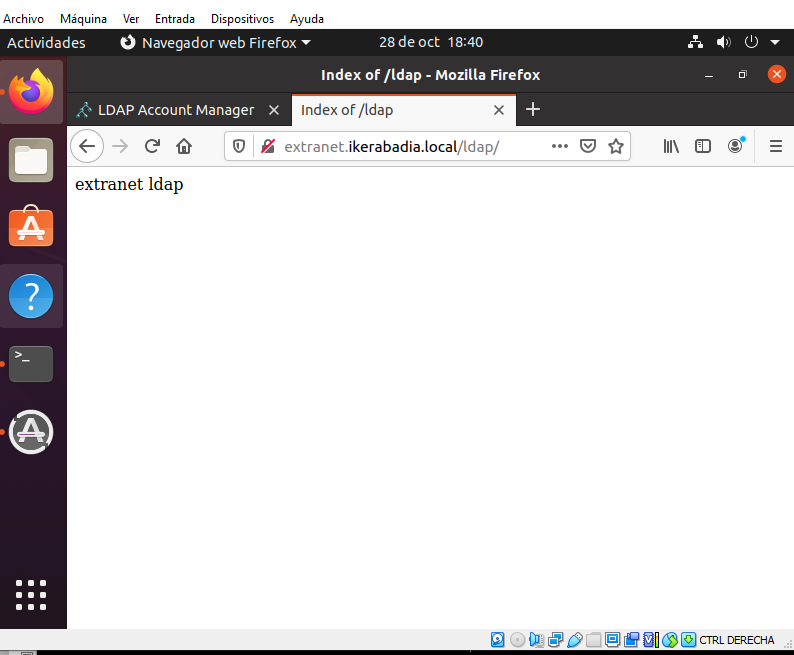


Cuando funcione le creo un index para que se vea al loguearme:



Reinicio

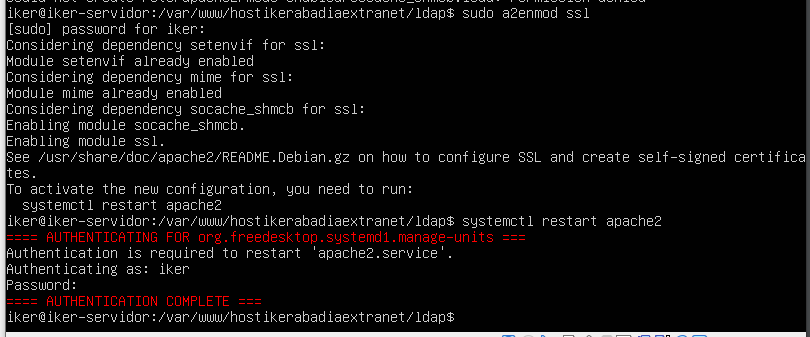




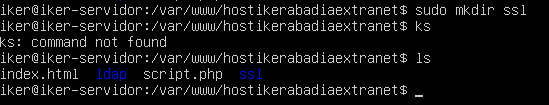
Perfecto

# G) Host Virtual seguro mediante HTTPS:

## a. Activamos el módulo ssl mediante: a2enmod ssl



## b. Creamos un directorio /ssl bajo /var/www/extranet/

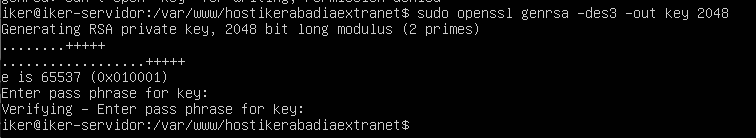


## c. Ubícate en el directorio anterior y genera los certificados:

**HAY QUE HACERLO EN LA CARPETA SSL NO EN LA DE IKERABADIAEXTRANET COMO SE VE EN LAS CAPTURAS**

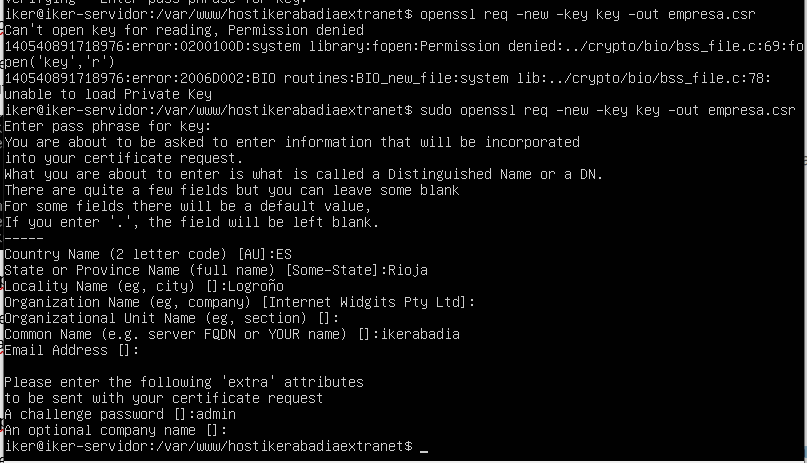
openssl genrsa -des3 -out key 2048

Genera un certificado clave SSL de 2048 bits de longitud. Pedirá un password para usarlo a futuro.



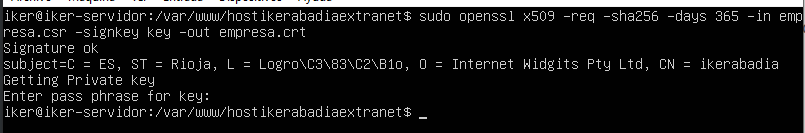
Openssl req -new -key key -out empresa.csr

Genera el certificado CSR (Certificate Signing Request) a partir de la clave anterior. Pedirá el password y una serie de campos



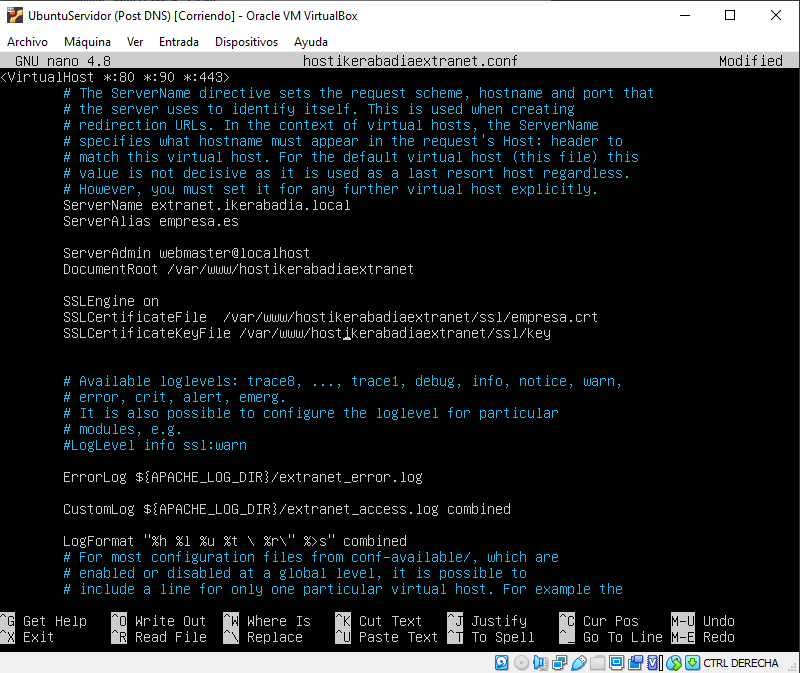
Openssl x509 -req -sha256 -days 365 -in empresa.csr -signkey key -out empresa.crt

Genera el certificado para nuestro sitio web. Pediré el password también.

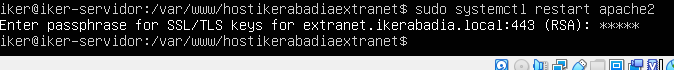


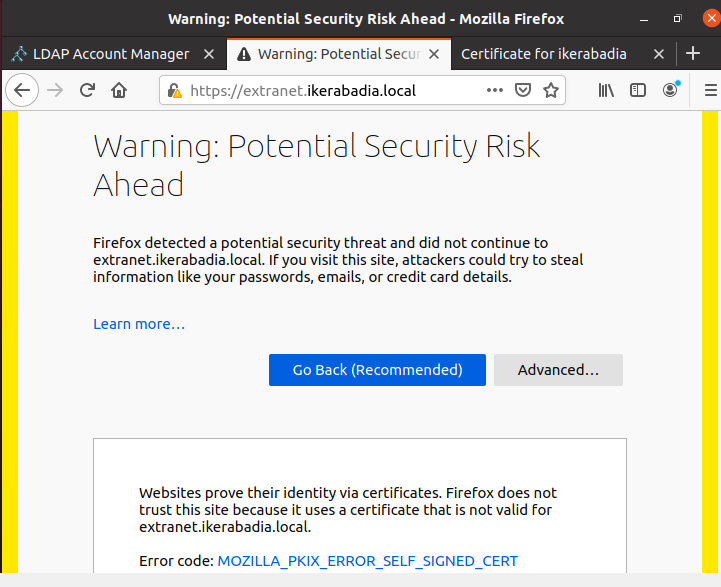
## d. Configura el host virtual análogamente a la siguiente captura, cambiando los parámetros necesarios (recuerda que a la extranet se puede acceder por dos puertos, ahora accederás por el tercero, el 443):



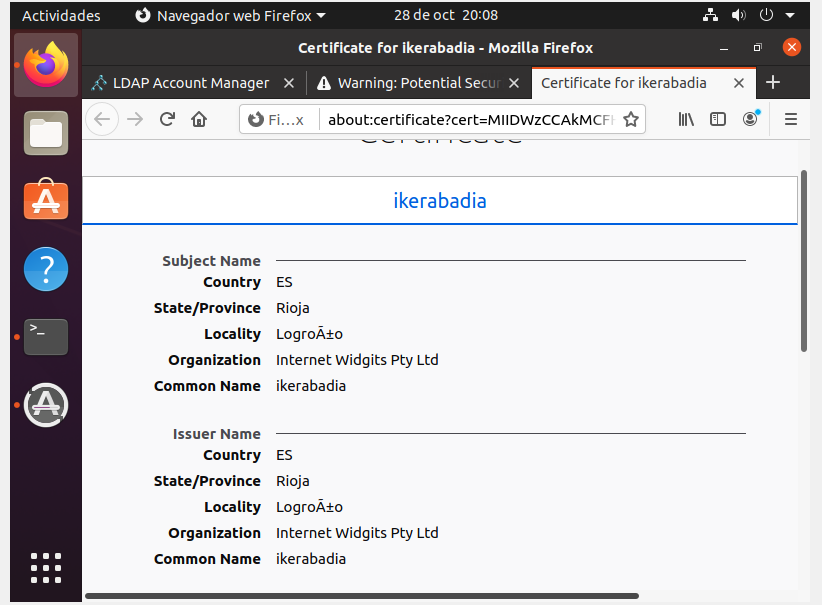


## e. Reinicia el servicio y verifica que el acceso a la extranet ya es seguro mediante https. Comprobarás que en Firefox se te avisa que aún así la conexión no es segura por ser un certificado autofirmado. Captura el certificado en el enlace que te ofrece el navegador.





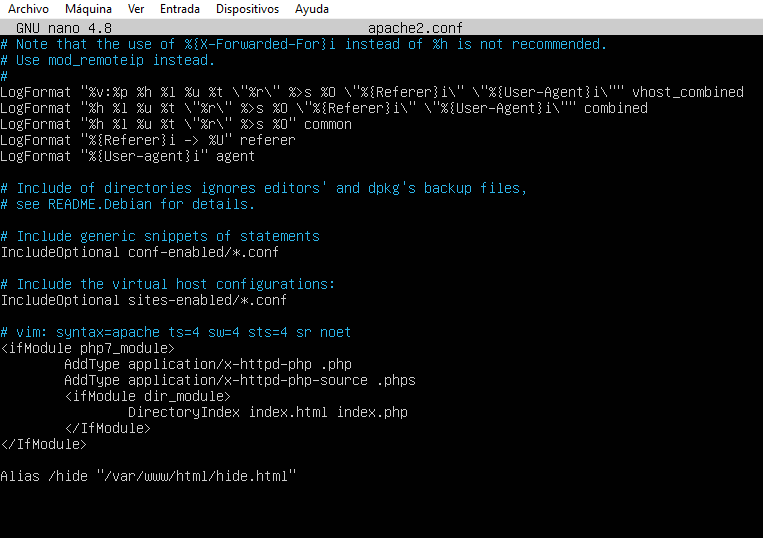
**NOTA: DEBES PONER EL HTTPS EN LA URL MANUALMENTE**



# H) Directorios y archivos de configuración:

## a. Genera un alias /oculto/ para una página web llamada hide.html ubicada en /var/www/hide/hide.html y demuestra que se renderiza vía URL en un navegador web.



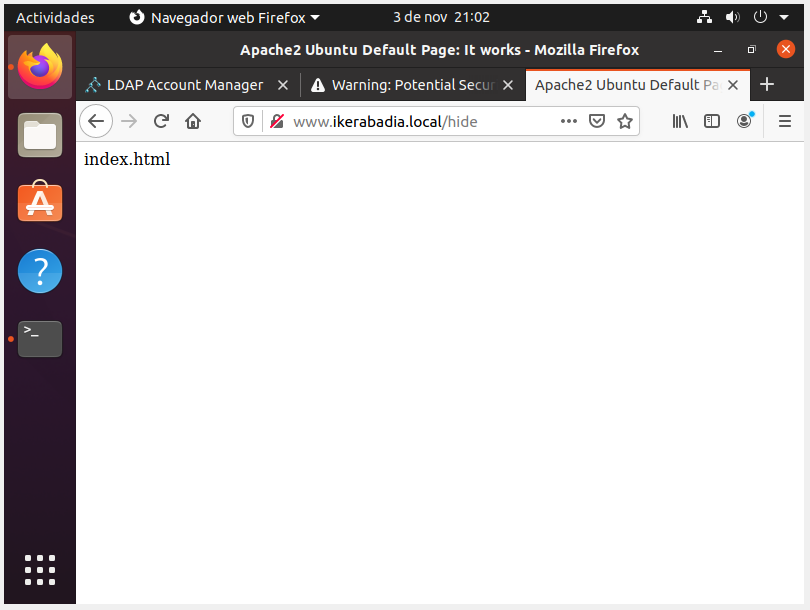


**ESTE ARCHIVO ES APACHE2.CONF:**

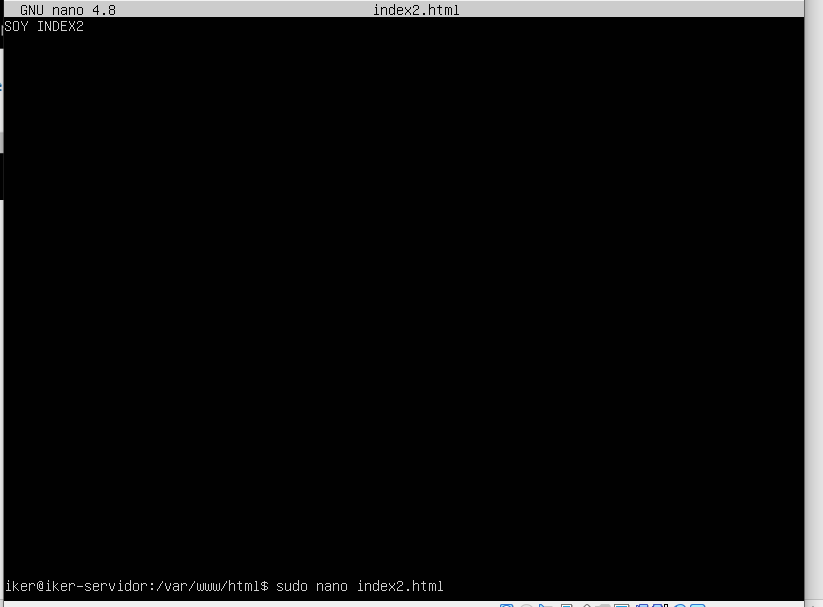


Reinicio:

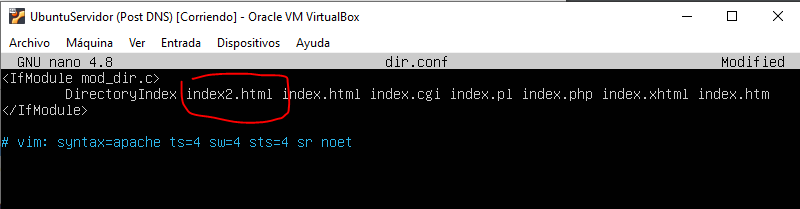




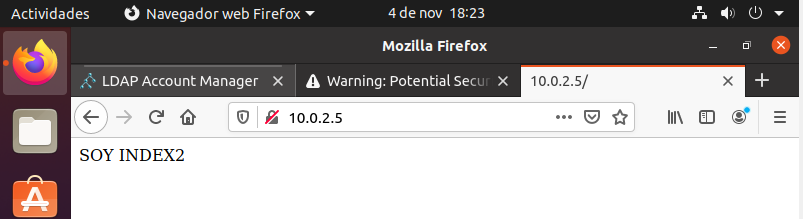
## b. Consigue el siguiente efecto (usa la directiva adecuada): crea el archivo web index2.html con el contenido: “SOY INDEX2” en el directorio de publicación principal y al indicar http://IP ó http://dominioprincipal debe salir directamente dicho contenido.

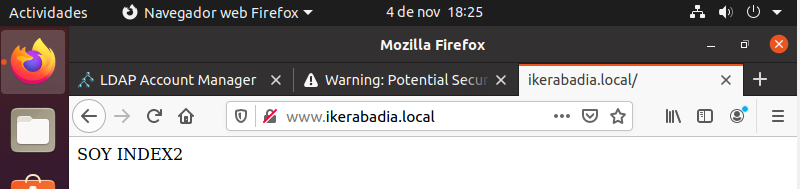






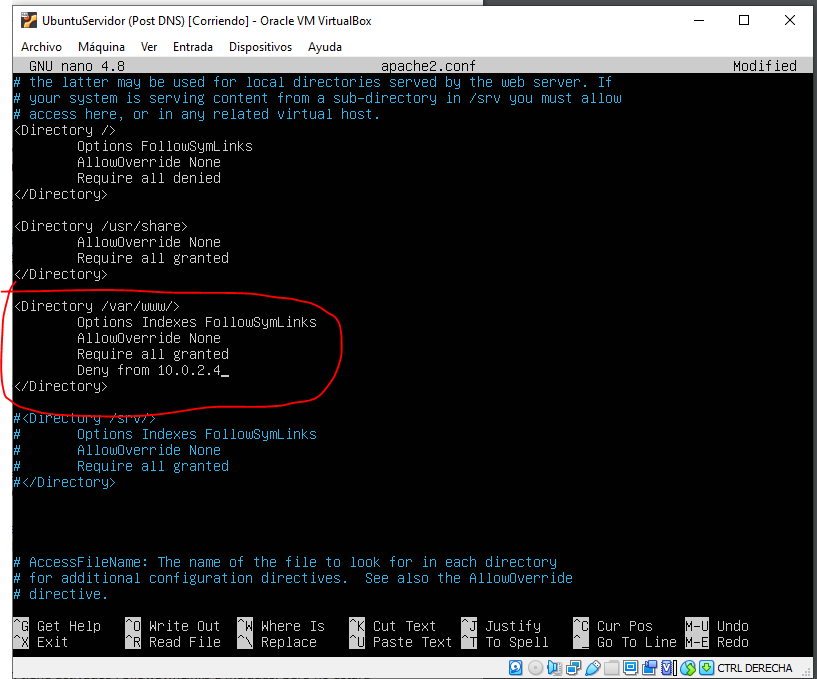


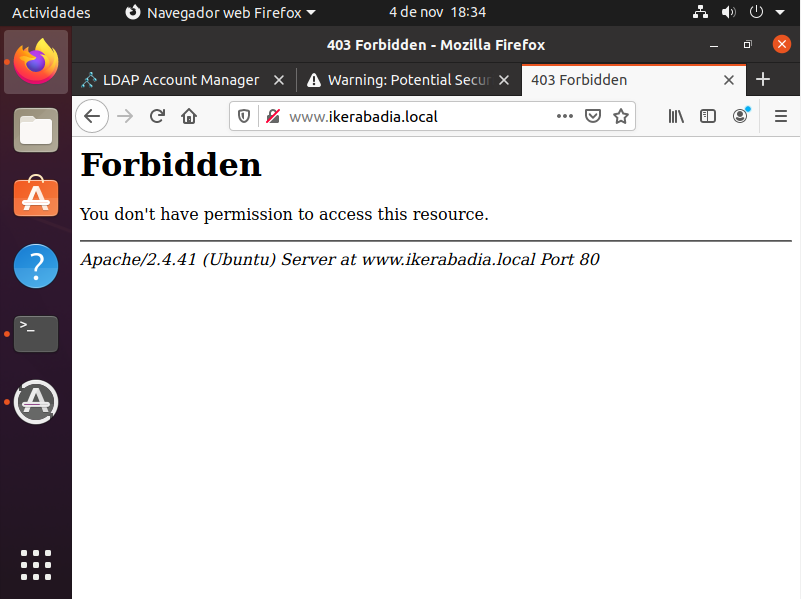




## c. Prohíbe el acceso al cliente basándote en su IP. Utiliza la directiva adecuada para el directory “/var/www”







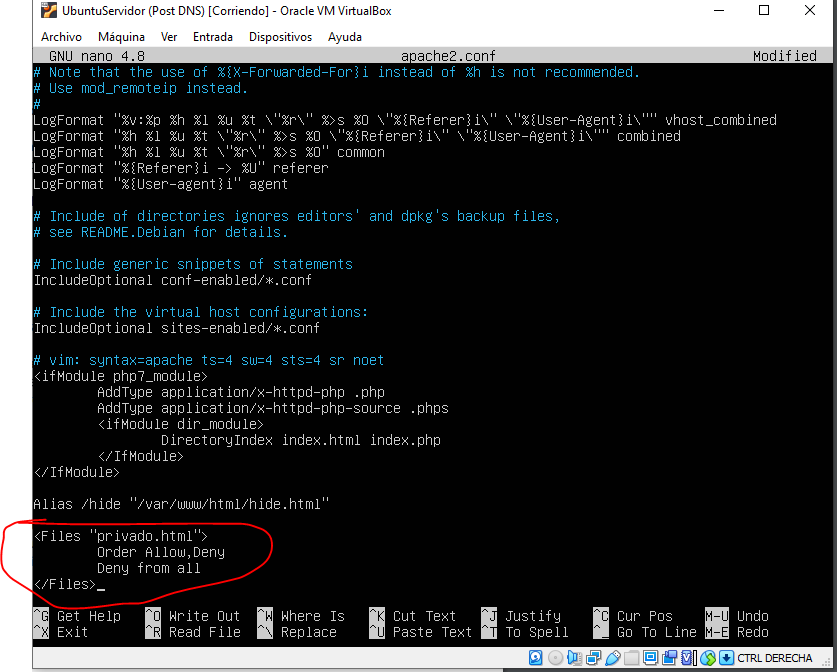
## d. Consigue que ningún archivo llamado privado.html pueda ser renderizado mediante en una directiva Files

Creo el archivo

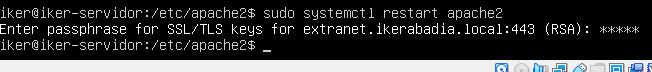


Añado la directiva a /etc/apache2/apache2.conf





Por último, reinicio el servicio





## e. /var/log/apache2/error.log -> Se registra cada solicitud a su servidor web en este archivo de registro, a menos que Apache esté configurado para hacer algo diferente. Verifica que dicho log crece con cada solicitud.

Ejecuta tail -f /var/log/apache2/\*.log



