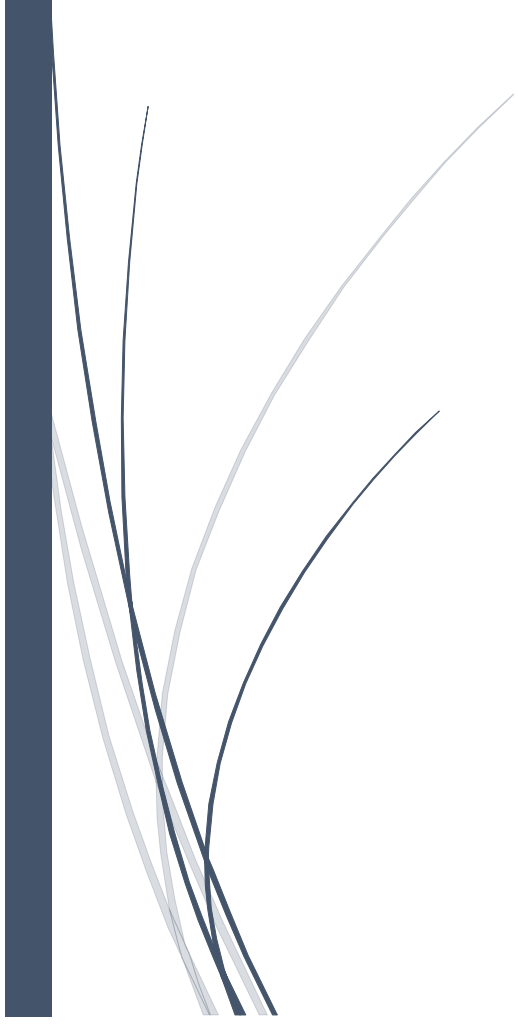
DESPLIEGUE DE APLICACIONES WEB

AUTENTICACIÓN EN NGINX



ÍNDICE

INTRODUCCIÓN

En esta práctica se va a aprender a hacer una autenticación básica que permita acceder a un sitio web a través de credenciales en forma de usuario y contraseña.

Una autenticación básica es la forma más simple de autenticación disponible para una aplicación web y no requiere de uso de cookies ni de identificadores de sesión ni de una página de login pero todo esto lleva aparejado una gran contra: la autenticación básica tiene un número importante de problemas de seguridad lo que hace que no sea recomendable su uso en la mayoría de las ocasiones.

Para ello necesitamos una máquina virtual que tenga instalado un SO Debian y el servidor web Nginx, en nuestro caso vamos a utilizar la máquina virtual de la práctica anterior donde se instaló el servidor web Nginx y el servidor de transferencia de ficheros SFTP:

Los paquetes que necesitamos tener instalados en nuestra máquina son los referentes a la herramienta openssl. Para comprobar que tenemos el paquete instalado podemos usar el siguiente comando:

sudo dpkg -l | grep openssl

Con este comando vamos a conseguir un listado de todos los paquetes instalados en el sistema (dpkg -l) y, posteriormente, vamos a filtrar con grep aquellos que tengan que ver con OpenSSL:

```
albertom-servidor@servidor-debian: $ sudo dpkg -l | grep openssl
ii  libxmlsec1-openssl:amd64          1.2.37-2                amd64        Openssl engine for the XML security library
ii  openssl                           3.0.9-1                 amd64        Secure Sockets Layer toolkit - cryptographic utility
ii  perl-openssl-defaults:amd64       7+b1                    amd64        version compatibility baseline for Perl OpenSSL packages
ii  python3-openssl                   23.0.0-1                all          Python 3 wrapper around the OpenSSL library
```

En el caso de que no tuviéramos estos paquetes instalados deberíamos instalarlos.

CREACIÓN DE USUARIOS Y CONTRASEÑAS PARA EL ACCESO WEB

El primer paso en esta práctica será la creación de dos usuarios con sus correspondientes contraseñas para poder acceder a un sitio web.

Para poder almacenar estas contraseñas en algún lugar vamos a crear un fichero oculto de nombre htpasswd dentro del directorio de configuración de nginx (/etc/nginx).

Usaremos el siguiente comando:

sudo sh -c "echo -n 'nombreUsuario:' >> /etc/nginx/.htpasswd"

```
albertom-servidor@servidor-debian:~$ sudo sh -c "echo -n 'alberto:' >> /etc/nginx/.htpasswd"
```

Ahora crearemos una contraseña cifrada para este usuario, con debemos usar el comando:

sudo sh -c "openssl passwd -apr1 >> /etc/nginx/.htpasswd"

Este comando lo que va a hacer es solicitarnos una contraseña que asociar al usuario y utilizar OpenSSL para generar una contraseña hash utilizando el algoritmo de cifrado APR1.

```
albertom-servidor@servidor-debian:~$ sudo sh -c "openssl passwd -apr1 >> /etc/nginx/.htpasswd"
Password:
Verifying - Password:
```

Repetimos el proceso con un segundo usuario:

```
albertom-servidor@servidor-debian:~$ sudo sh -c "echo -n 'martinez:' >> /etc/nginx/.htpasswd"
albertom-servidor@servidor-debian:~$ sudo sh -c "openssl passwd -apr1 >> /etc/nginx/.htpasswd"
Password:
Verifying - Password:
```

Si hacemos un cat del archivo .htpasswd veremos lo siguiente:

```
albertom-servidor@servidor-debian:~$ sudo cat /etc/nginx/.htpasswd
alberto:$apr1$45Q4yLgx$bQTiRddRWuYHys4jc68fD.
martinez:$apr1$jTflj.ob$bL9Gil7pYsFXPUrbYGWrO/
```

CONFIGURACIÓN DE NGINX PARA EL USUARIO DE LA AUTENTICACIÓN BÁSICA

Vamos a editar la configuración del server block del sitio web www.academia.es creado en el último punto de la práctica anterior.

Para ello abrimos el archivo de configuración del sitio (alojado en /etc/nginx/sites-available/www.academia.es) utilizando nano:

sudo nano /etc/nginx/sites-available/www.academia.es

```
albertom-servidor@servidor-debian:~$ sudo nano /etc/nginx/sites-available/www.academia.es
```

Nginx nos permite añadir restricciones a nivel de servidor o en una localización en específico (puede ser un directorio o un archivo), en esta práctica vamos a proteger el documento root (el home del sitio).

Para conseguir esto vamos a usar dos directivas:

- **auth_basic**: Define el mensaje que se le mostrará al usuario al intentar entrar a la zona, que hemos limitado.
- **auth_basic_user_file**: En esta directiva vamos a especificar dónde se encuentra la información de autenticación (en nuestro caso será el archivo oculto .htpasswd que hemos creado en la sección anterior).

Como vamos a afectar al documento root, estas directivas se incluirán en el sub-bloque "location /":

```
location / {
        auth_basic "Área restringida";
        auth_basic_user_file /etc/nginx/.htpasswd;
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
}
```

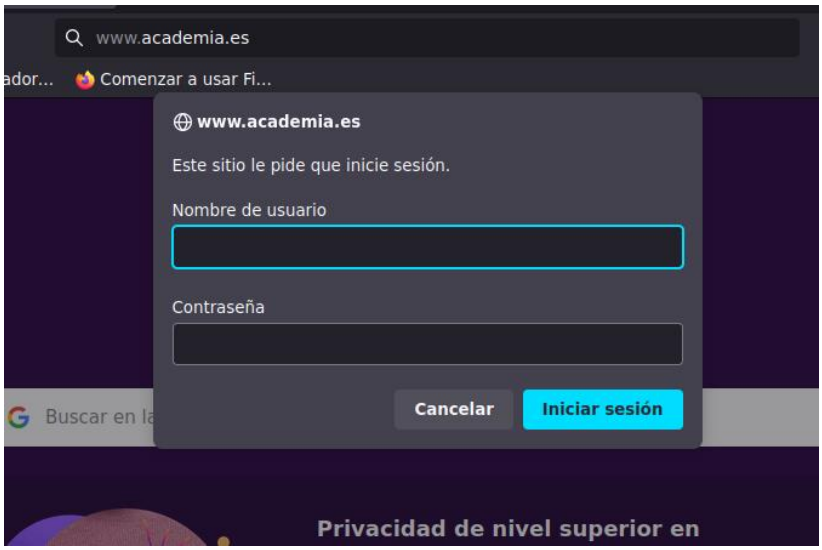
Como hemos cambiado la configuración de nuestro servidor debemos hacer un reinicio de este y comprobar posteriormente su estado, para ello utilizaremos las opciones restart y status del comando systemctl:

```
albertom-servidor@servidor-debian:~$ sudo systemctl restart nginx.service
albertom-servidor@servidor-debian:~$ sudo systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Fri 2023-10-20 17:05:07 CEST; 5s ago
       Docs: man:nginx(8)
    Process: 1484 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 1485 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 1486 (nginx)
      Tasks: 3 (limit: 2284)
     Memory: 2.3M
        CPU: 22ms
     CGroup: /system.slice/nginx.service
             ├─1486 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─1487 "nginx: worker process"
             └─1488 "nginx: worker process"

oct 20 17:05:07 servidor-debian systemd[1]: Stopped nginx.service - A high performance web server and a reverse prox>
oct 20 17:05:07 servidor-debian systemd[1]: Starting nginx.service - A high performance web server and a reverse pro>
oct 20 17:05:07 servidor-debian systemd[1]: Started nginx.service - A high performance web server and a reverse prox>
```

COMPROBAR LA AUTENTICACIÓN

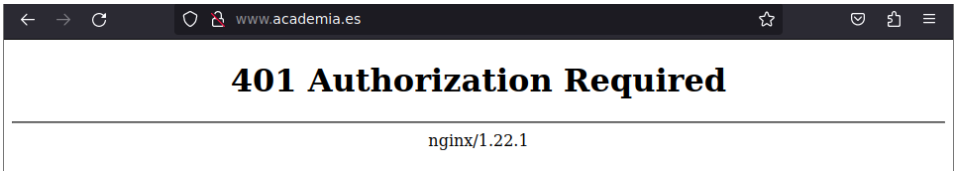
Si todos los pasos anteriores se han completado exitosamente y de forma correcta, si ahora intentamos acceder al sitio web www.academia.es deberíamos ver la siguiente pantalla:



Si nos autenticamos de forma incorrecta el sitio web nos seguirá solicitando el introducir la contraseña, en caso de que introduzcamos la contraseña correcta podremos navegar sin problemas:

Hay que tener en cuenta que una vez que nos autentiquemos por primera vez esto quedará almacenado en el navegador y no volverá a solicitar usuario y contraseña al entrar al sitio. Para volver a probar la autenticación deberemos usar una ventana privada en el navegador.

En el caso de que se pulse cancelar en el modal que solicita usuario y contraseña aparecerá la siguiente información en pantalla:



REGISTRO DE SUCESOS EN LOS ARCHIVOS .log

Ahora podemos verificar estos intentos de conexión en los archivos .log de nginx que se comentaron en la práctica anterior.

Vamos a visualizar primero el log de accesos:

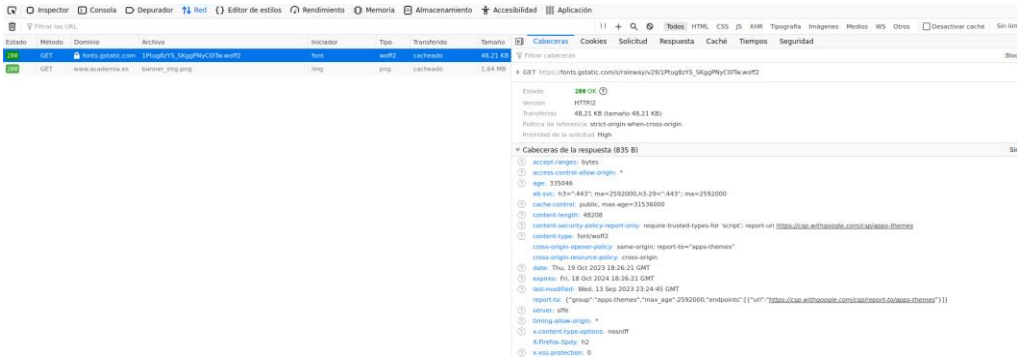
sudo tail /var/log/nginx/access.log

```
albertom-servidor@servidor-debian:~$ sudo tail /var/log/nginx/access.log
10.0.2.20 - alberto [23/Oct/2023:17:27:32 +0200] "GET /css/magnific-popup.css HTTP/1.1" 200
10.0.2.20 - alberto [23/Oct/2023:17:27:32 +0200] "GET /css/responsiveslides.css HTTP/1.1" 2
10.0.2.20 - alberto [23/Oct/2023:17:27:32 +0200] "GET /css/timeline.css HTTP/1.1" 200 75853
10.0.2.20 - alberto [23/Oct/2023:17:27:32 +0200] "GET /css/flaticon.css HTTP/1.1" 200 908 "
10.0.2.20 - alberto [23/Oct/2023:17:27:33 +0200] "GET /images/heading_main_border.png HTTP/
10.0.2.20 - alberto [23/Oct/2023:17:27:33 +0200] "GET /images/banner_img.png HTTP/1.1" 200
10.0.2.20 - alberto [23/Oct/2023:17:27:33 +0200] "GET /fonts/fontawesome-webfont.woff2?v=4.
```

En este archivo podemos ver los accesos exitosos al sitio web, así como las peticiones que se han hecho al sitio web.

En cada línea tenemos la IP de conexión, el usuario, la fecha, el recurso solicitado, el navegador utilizado, etc.

Esta información es visible también desde la herramientas de desarrollador del navegador:



Podemos ver también el archivo log de errores:

sudo tail /var/log/nginx/error.log

```
albertom-servidor@servidor-debian:~$ sudo tail /var/log/nginx/error.log
2023/10/23 17:27:51 [error] 1406#1406: *7 user "alberto": password mismatch, client: 10.0.2.20,
```

En este caso vemos las conexiones erróneas que hemos provocado con el usuario 'alberto' (cuando ha intentado conectarse pero la clave era errónea).

RESTRICCIÓN DE ACCESO POR IP

Para ver funcionar esto de forma correcta vamos a usar nuestra máquina cliente con un Ubuntu 22.04 instalado.

Como no contamos con un DNS para nuestras webs www.estatica.es y www.academia.es debemos ir al archivo /etc/hosts y añadir estas líneas:

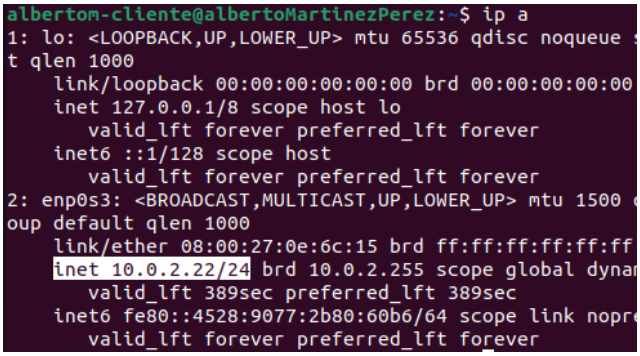
```
  GNU nano 6.2                          /etc/hosts *
127.0.0.1       localhost
127.0.1.1       albertomcliente-VirtualBox
10.0.2.20       www.estatica.es
10.0.2.20       www.academia.es
```

Siendo la dirección IP aquella que tengamos asignada en nuestra maquina servidora Debian.

Este tipo de restricción de acceso al sitio web se puede aplicar de dos formas:

A. DENEGACIÓN DE ACCESO SOBRE IP CONCRETA

Vamos a negar el acceso a nuestra máquina virtual Ubuntu sobre el sitio web www.academia.es, para ello debemos primero conocer la IP de nuestra máquina:



En nuestro servidor, escribir el siguiente código en el location /:

```
location / {
        deny 10.0.2.22;
        allow 10.0.2.0/24;
        allow 127.0.0.1;
        deny all;
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
}
```

Esto significa que se denegará el acceso a la dirección IP 10.0.2.22, se permitirá al resto de las direcciones de la red 10.0.2.0/24.

La última orden implica denegar el acceso a todo el mundo. Estas ordenes se ejecutan de arriba abajo, por eso hay que poner allow y deny específicos antes de esta última directiva.

Si ahora intentamos acceder al sitio web recibiremos el siguiente mensaje:

Si investigamos en los logs podremos ver lo siguiente:

```
albertom-servidor@servidor-debian:~$ sudo tail /var/log/nginx/error.log
2023/10/23 17:27:51 [error] 1406#1406: *7 user "alberto": password mismatch, client: 1
es, request: "GET / HTTP/1.1", host: "www.academia.es"
2023/10/23 17:58:13 [error] 2209#2209: *4 access forbidden by rule, client: 10.0.2.22,
st: "GET / HTTP/1.1", host: "www.academia.es"
2023/10/23 17:58:14 [error] 2209#2209: *4 access forbidden by rule, client: 10.0.2.22,
st: "GET /favicon.ico HTTP/1.1", host: "www.academia.es", referrer: "http://www.academ
```

Se ha registrado este intento de acceso a la página como un error de tipo "access forbidden by rule".

B. COMBINACIÓN DE RESTRICCIÓN IP Y AUTENTICACIÓN HTTP

Para unir ambas formas de permiso de acceso al sitio web vamos a utilizar la directiva satisfy. Esta directiva admite dos posibles valores:

a) **any**: Permite que el cliente acceda si cumple uno de los requisitos.

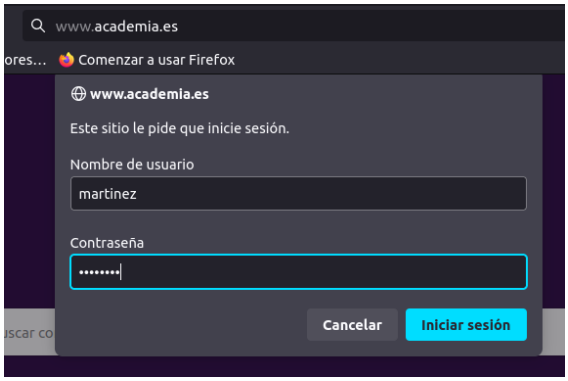
```
location / {
        satisfy any;

        deny 10.0.2.22;
        allow 10.0.2.0/24;
        allow 127.0.0.1;
        deny all;

        auth_basic "Area restringida";
        auth_basic_user_file /etc/nginx/.htpasswd;
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
}
```

En este caso aunque nuestra IP sigue estando denegada, tenemos la posibilidad de autenticarnos con uno de los usuarios y contraseña que creamos previamente.

Realizamos un restart del servicio Nginx y probamos a acceder a la web en nuestra máquina Ubuntu:



Si completamos bien los campos usuario y contraseña podremos acceder al sitio sin problemas.

Para comprobar que el acceso podemos, de nuevo, comprobar el log de acceso al sitio:

```
albertom-servidor@servidor-debian: $ sudo tail /var/log/nginx/access.log
10.0.2.22 - martinez [23/Oct/2023:18:01:15 +0200] "GET /images/footer_logo.png HTTP/1.1" 200 5737 "http://w
a.es/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - martinez [23/Oct/2023:18:01:15 +0200] "GET /css/animate.css HTTP/1.1" 200 56693 "http://www.aca
ss/style.css" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - martinez [23/Oct/2023:18:01:15 +0200] "GET /css/font-awesome.min.css HTTP/1.1" 200 31000 "http:
emia.es/css/style.css" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
```

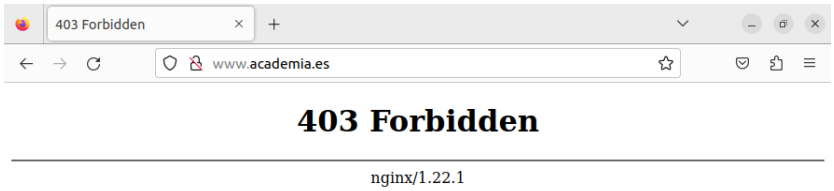
b) **all**: En este caso el cliente debe satisfacer todas las condiciones que se expresen más abajo.

```
location / {
        satisfy all;

        deny 10.0.2.22;
        allow 10.0.2.0/24;
        allow 127.0.0.1;
        deny all;

        auth_basic "Area restringida";
        auth_basic_user_file /etc/nginx/.htpasswd;
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
}
```

En este caso como la IP de nuestra máquina sigue con acceso denegado volveremos a recibir el mensaje 403 forbidden al intentar acceder al sitio:



Para comprobar este acceso fallido debemos acudir al log de errores:

```
albertom-servidor@servidor-debian:~$ sudo tail /var/log/nginx/error.log
2023/10/23 17:27:51 [error] 1406#1406: *7 user "alberto": password mismatch, client: 10.0.2.20, s
es, request: "GET / HTTP/1.1", host: "www.academia.es"
2023/10/23 17:58:13 [error] 2209#2209: *4 access forbidden by rule, client: 10.0.2.22, server: ww
st: "GET / HTTP/1.1", host: "www.academia.es"
2023/10/23 17:58:14 [error] 2209#2209: *4 access forbidden by rule, client: 10.0.2.22, server: ww
st: "GET /favicon.ico HTTP/1.1", host: "www.academia.es", referrer: "http://www.academia.es/"
2023/10/23 18:04:54 [error] 2277#2277: *2 access forbidden by rule, client: 10.0.2.22, server: ww
st: "GET / HTTP/1.1", host: "www.academia.es"
2023/10/23 18:04:54 [error] 2277#2277: *2 access forbidden by rule, client: 10.0.2.22, server: ww
st: "GET /favicon.ico HTTP/1.1", host: "www.academia.es", referrer: "http://www.academia.es/"
```

RESTRICCIÓN DE ACCESO SOBRE UNA LOCALIZACIÓN EN ESPECÍFICO DEL SITIO WEB

Hemos hecho en pasos anteriores una restricción de acceso sobre el raíz de nuestro sitio web lo que afectaba al home, ahora vamos a hacer una restricción sobre una sección en especifico de la web, por ejemplo, la página de contacto (contact.html).

Para conocer la ruta de esta página podemos hacer por ejemplo un tree del directorio /var/www/fich_academia/:

tree /var/www/fich_academia/

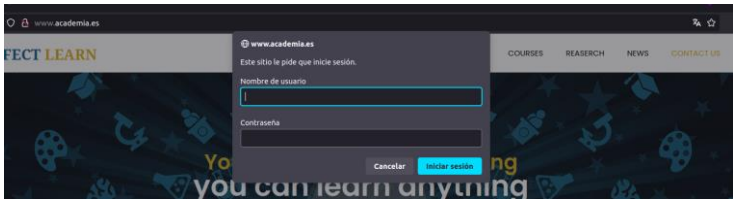
```
albertom-servidor@servidor-debian:~$ sudo tree /var/www/fich_academia/
/var/www/fich_academia/
├── about.html
├── contact.html
├── css
│   ├── animate.css
│   ├── bootstrap.min.css
│   ├── custom.css
│   ├── flashy.min.css
│   ├── flaticon.css
│   ├── font-awesome.min.css
│   ├── magnific-popup.css
│   ├── pogo-slider.min.css
│   ├── responsive.css
│   ├── responsiveslides.css
│   ├── style.css
│   └── timeline.css
├── fonts
│   ├── Flaticon.eot
│   ├── Flaticon.svg
│   ├── Flaticon.ttf
│   ├── Flaticon.woff
│   ├── FontAwesome.otf
│   ├── fontawesome-webfont.eot
│   ├── fontawesome-webfont.svg
│   ├── fontawesome-webfont.ttf
│   ├── fontawesome-webfont.woff
│   └── fontawesome-webfont.woff2
```

Por lo que se puede ver el archivo contact.html pende directamente del raíz por lo que la dirección del archivo será "/contact.html". Para conseguir bloquear el acceso a este sitio sin autenticación deberemos crear la siguiente regla de localización:

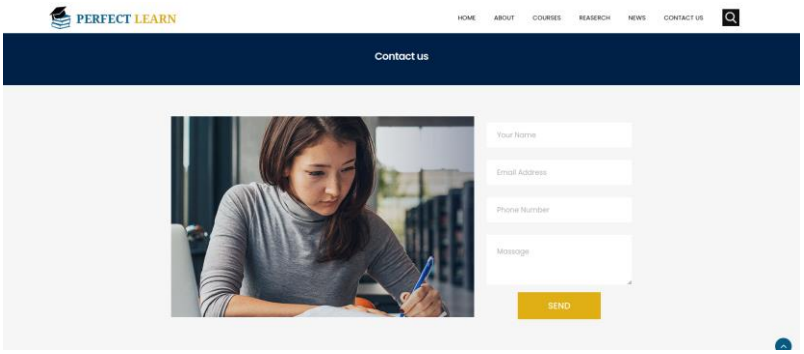
```
location /contact.html {
        auth_basic              "Area restringida";
        auth_basic_user_file    /etc/nginx/.htpasswd;
        try_files $uri $uri/ =404;
}
```

Ahora si intentamos acceder a nuestro sitio web lo podremos hacer sin problemas:

Pero en el caso de querer acceder a la sección "contact us" se nos solicitará introducir nuestro usuario y contraseña:



Si completamos correctamente el formulario de inicio de sesión podremos acceder a la sección:



Esta acción como el resto de las que hemos visto en esta guía queda archivada en los logs de acceso del servidor:

```
albertom-servidor@servidor-debian:~$ sudo tail /var/log/nginx/access.log
10.0.2.22 - alberto [23/Oct/2023:18:13:45 +0200] "GET /js/bootstrap.min.js HTTP/1.1" 304 0 "http://www.academia.es/contact.html" "M
; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - alberto [23/Oct/2023:18:13:45 +0200] "GET /js/jquery.magnific-popup.min.js HTTP/1.1" 304 0 "http://www.academia.es/cont
lla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - alberto [23/Oct/2023:18:13:45 +0200] "GET /js/jquery.pogo-slider.min.js HTTP/1.1" 304 0 "http://www.academia.es/contact
/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - alberto [23/Oct/2023:18:13:45 +0200] "GET /js/slider-index.js HTTP/1.1" 304 0 "http://www.academia.es/contact.html" "Mo
 Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - alberto [23/Oct/2023:18:13:45 +0200] "GET /js/smoothscroll.js HTTP/1.1" 304 0 "http://www.academia.es/contact.html" "Mo
 Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - alberto [23/Oct/2023:18:13:45 +0200] "GET /js/form-validator.min.js HTTP/1.1" 304 0 "http://www.academia.es/contact.htm
 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - alberto [23/Oct/2023:18:13:45 +0200] "GET /js/contact-form-script.js HTTP/1.1" 304 0 "http://www.academia.es/contact.ht
0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - alberto [23/Oct/2023:18:13:45 +0200] "GET /js/isotope.min.js HTTP/1.1" 304 0 "http://www.academia.es/contact.html" "Moz
Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - alberto [23/Oct/2023:18:13:45 +0200] "GET /js/images-loded.min.js HTTP/1.1" 304 0 "http://www.academia.es/contact.html"
X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - alberto [23/Oct/2023:18:13:45 +0200] "GET /js/custom.js HTTP/1.1" 304 0 "http://www.academia.es/contact.html" "Mozilla/
u; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
```

CUESTIONES FINALES

1. Supongamos que yo soy el cliente con la IP 172.1.10.15 e intento acceder al directorio **web_muy_guay** de mi sitio web, equivocándome al poner el usuario y contraseña. ¿Podré acceder? ¿Por qué?

```
    location /web_muy_guay {
    #...
    satisfy all;
    deny  172.1.10.6;
    allow 172.1.10.15;
    allow 172.1.3.14;
    deny  all;
    auth_basic "Cuestión final 1";
    auth_basic_user_file conf/htpasswd;
}
```

   No podrás acceder por las siguientes razones:

   a. Para acceder al sitio debes satisfacer tanto la restricción de IP como la autenticación usuario-contraseña.

b. Aunque tu IP tiene la directiva allow asignada y te permitiría acceder, has introducido mal tu autenticación y por tanto no puedes acceder.

2. Supongamos que yo soy el cliente con la IP 172.1.10.15 e intento acceder al directorio web_muy_guay de mi sitio web, introduciendo correctamente usuario y contraseña. ¿Podré acceder? ¿Por qué?

```
location /web_muy_guay {
#...
satisfy all;
deny  all;
deny  172.1.10.6;
allow 172.1.10.15;
allow 172.1.3.14;

auth_basic "Cuestión final 2: The revenge";
auth_basic_user_file conf/htpasswd;
}
```

No podrás acceder por las siguientes razones:

a. Para acceder debes satisfacer tanto la restricción de IP como la de autenticación.
b. Aunque tu autenticación usuario-contraseña ha sido correcta, la cascada de permisos deny/allow comienza con un deny all por lo que se denegará el acceso a todos los usuarios que intenten acceder a esta localización.

3. Supongamos que yo soy el cliente con la IP 172.1.10.15 e intento acceder al directorio **web_muy_guay** de mi sitio web, introduciendo correctamente usuario y contraseña. ¿Podré acceder? ¿Por qué?

```
location /web_muy_guay {
#...
satisfy any;
deny  172.1.10.6;
deny 172.1.10.15;
allow 172.1.3.14;

auth_basic "Cuestión final 3: The final combat";
auth_basic_user_file conf/htpasswd;
}
```

Podrás acceder al sitio porque:

a. Para acceder debes satisfacer una de las dos formas o bien la restricción de IP o bien la autenticación.
b. Tu IP se encuentra restringida porque tiene la directiva deny asignada pero como te has autenticado de forma correcta con tu usuario y contraseña, se te concederá el acceso.

4. A lo mejor no sabéis que tengo una web para documentar todas mis excursiones espaciales con Jeff, es esta: Jeff Bezos y yo.

   Supongamos que quiero restringir el acceso al directorio de proyectos porque es muy secreto, eso quiere decir añadir autenticación básica a la URL:Proyectos.

   Completa la configuración para conseguirlo:

Habría que completar esta location:

```
location /projects.html {
        auth_basic  “Area restringida proyectos”;
        auth_basic_user_file /etc/nginx/.htpasswd;
        try_files $uri $uri/ =404;
}
```

Nota: Supongo que el archive donde guardo usuarios y contraseñas se llama .htpasswd y se aloja en el directorio /etc/nginx/.