



# APACHE SERVER

MECANISMO DE AUTENTICACIÓN BASIC

Salvador Mira

IES Pere Maria Orts

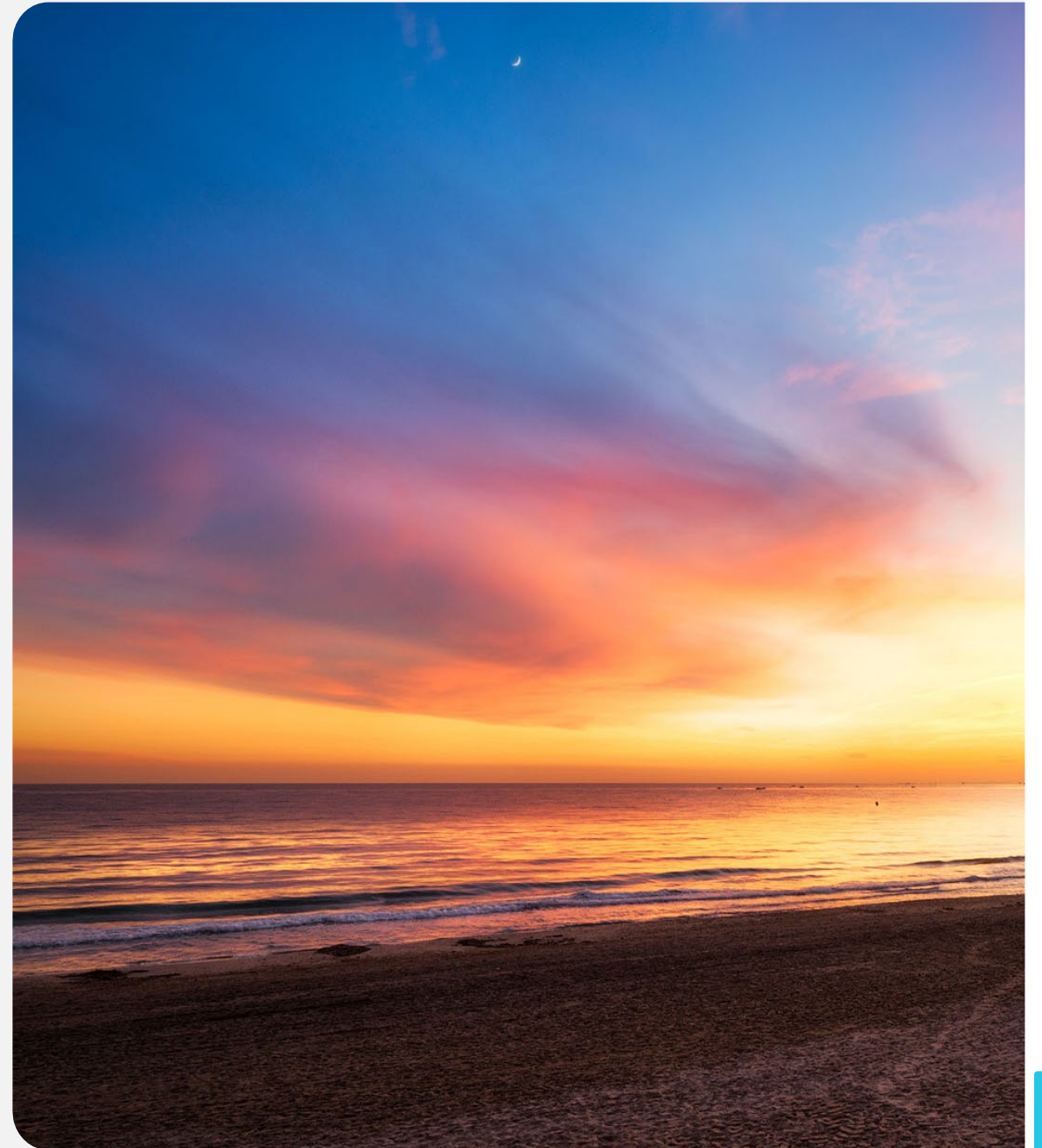
# MECANISMOS DE AUTENTICACIÓN

Podemos restringir el acceso a recursos mediante usuarios y contraseñas

Es posible organizar los usuarios en grupos y otorgar permisos de grupo

**IMPORTANTE:** Los mecanismos de Apache no incorporan ningún tipo de encriptación (sólo alguna codificación) entre cliente y servidor

- La encriptación se consigue con SSL o TLS
- En caso contrario el usuario y la contraseña se transmitirán entre el cliente y el servidor en texto plano
- Los mecanismos de encriptación de Apache se aplican solo al fichero en el que se almacenan las contraseñas





# MECANISMOS DE AUTENTICACIÓN

## FUNCIONAMIENTO

- Cuando se trata de acceder a un recurso protegido, Apache devuelve un mensaje **401, Authentication Required**
- Es posible que el navegador tenga las credenciales guardadas, y en tal caso las envía directamente
- Si el navegador no tiene las credenciales, las solicita al usuario
- Hay dos mecanismos en servidor
  - Autenticación Basic
  - Autenticación Digest



# AUTHENTICATION BASIC

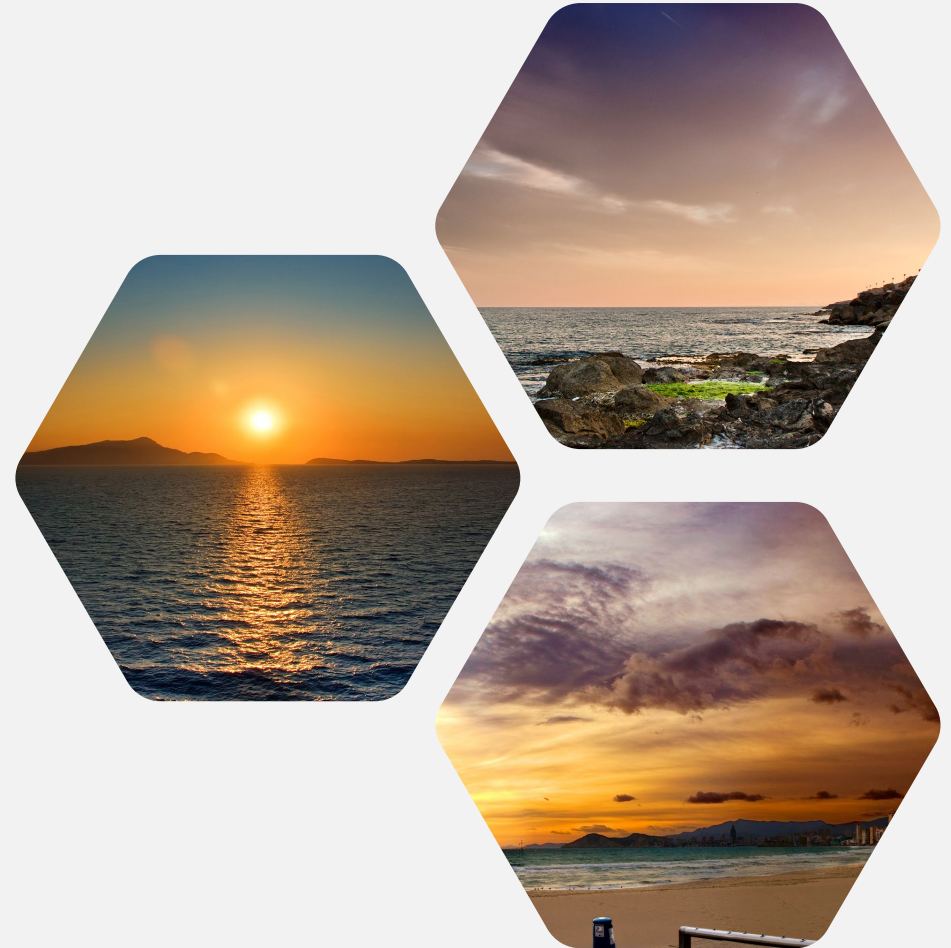
Lo primero que se necesita es un archivo en el que se almacenen las contraseñas

Podemos configurar la autenticación **basic** para que utilice distintos **proveedores de información**

- El más simple consiste en leer las credenciales de un fichero de texto

Si hacemos uso de un fichero de texto, éste debería estar en un lugar no accesible desde la web

- Por ejemplo en `/usr/local/apache/passwd/`



# AUTHENTICATION BASIC

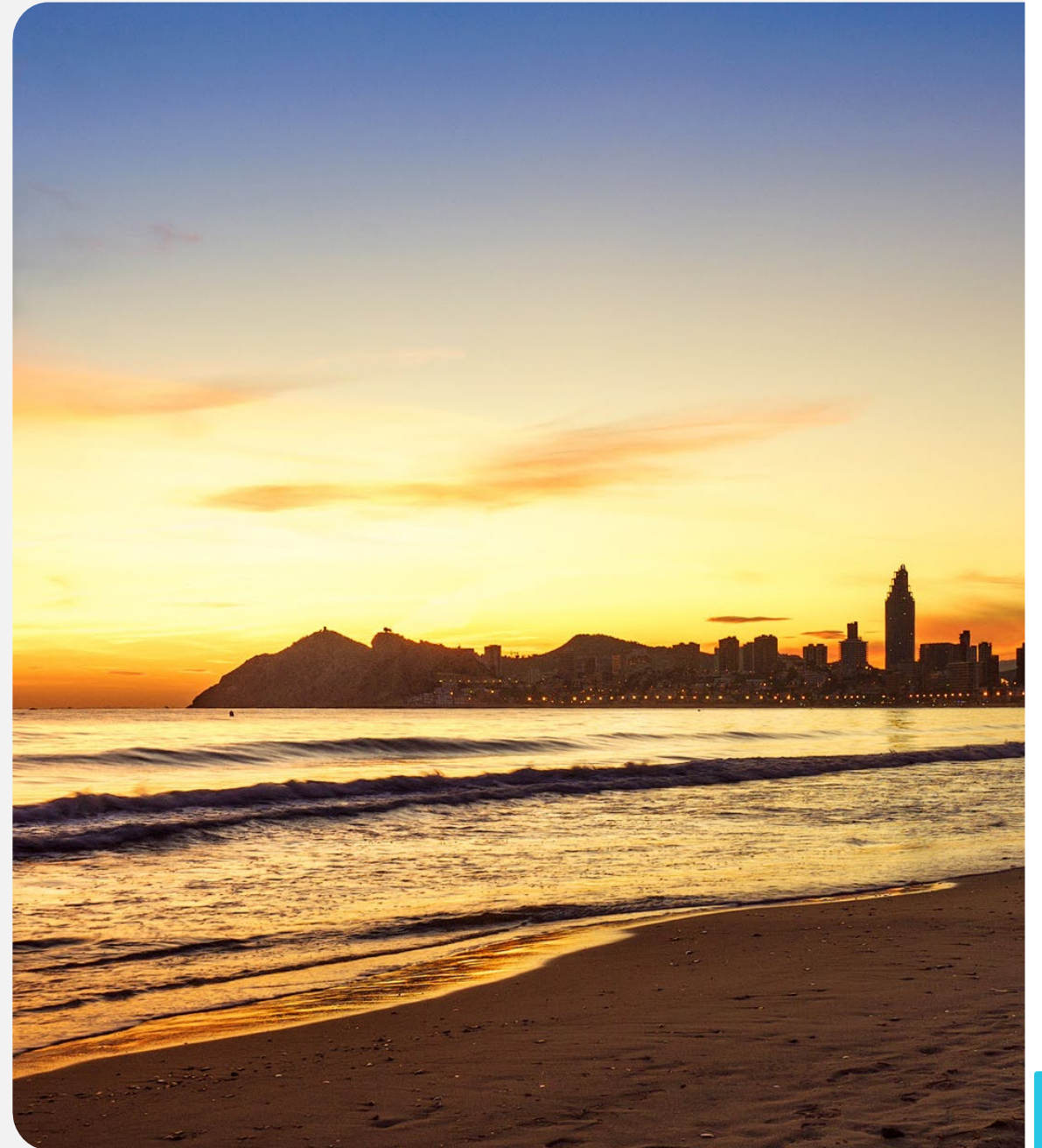
Para crear los usuarios y contraseñas debemos usar la utilidad **htpasswd** de Apache

Creación de passwords:

- `htpasswd -c /ruta/al/fichero/de/passwords usuario`

Añadir **nuevo usuario** a un fichero de contraseñas existente:

- `htpasswd /ruta/al/fichero/de/passwords usuario`





# AUTHENTICATION BASIC

```
administrador@debian-despliegue: /usr/local/apache/passwd
administrador@debian-despliegue:/usr/local/apache/passwd$ sudo htpasswd -c ./passwords administrador
New password:
Re-type new password:
Adding password for user administrador
administrador@debian-despliegue:/usr/local/apache/passwd$ sudo htpasswd ./passwords alumno
New password:
Re-type new password:
Adding password for user alumno
administrador@debian-despliegue:/usr/local/apache/passwd$ cat passwords
administrador:$apr1$y4Yw1D8C$HBuF36q7jmtEfeDJ8ww6L/
alumno:$apr1$.GZAHRkG$3oa2AXEUEoBnQ5cQ3Sc5i.
administrador@debian-despliegue:/usr/local/apache/passwd$ |
```

# AUTHENTICATION BASIC

Tras crear el fichero de usuarios y contraseñas, hay que definir un recurso protegido, habitualmente un directorio

En tal recurso hay que indicar, al menos, las siguientes directivas:

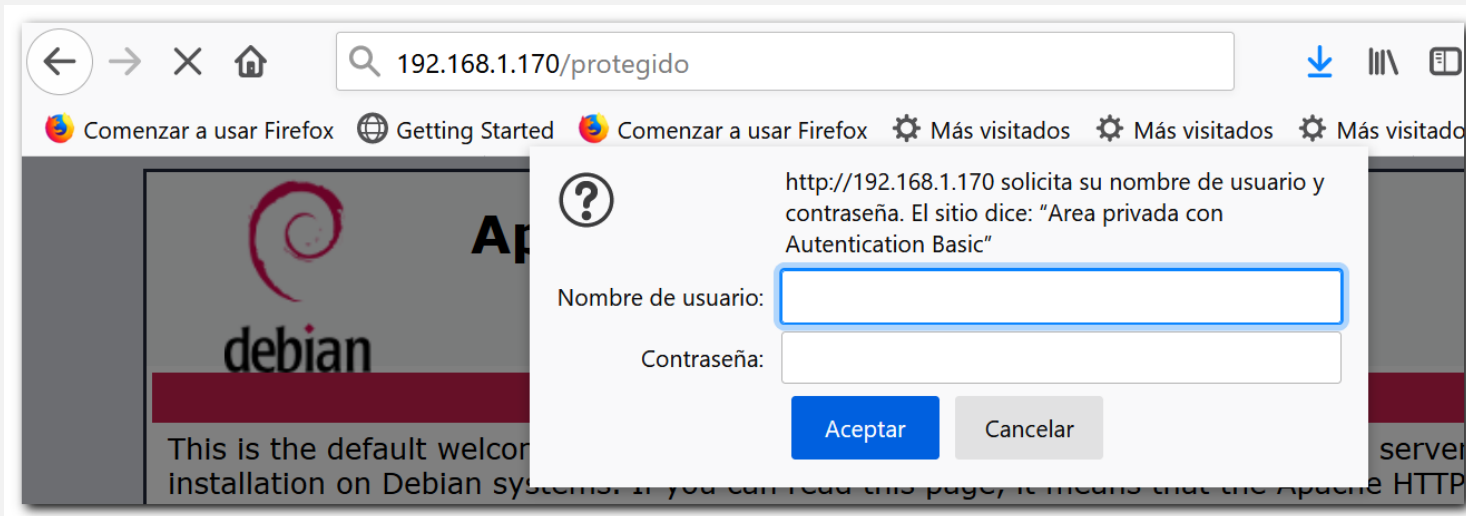
- `AuthType Basic`
- `AuthName "Area privada"` (puede ser cualquier texto)
- `AuthBasicProvider file`
- `AuthUserFile /ruta/al/fichero/de/passwords`
- `require valid-user` o bien  
`require user usuario1 usuario2`



# AUTHENTICATION BASIC

```
<Directory "/var/www/html/protegido">  
    AuthType Basic  
    require valid-user  
    AuthBasicProvider file  
    AuthUserFile "/usr/local/apache/passwd/password"  
    AuthName "Area privada con Authentication Basic"  
</Directory>
```

Definición de directorio protegido en el VirtualHost por defecto de Apache

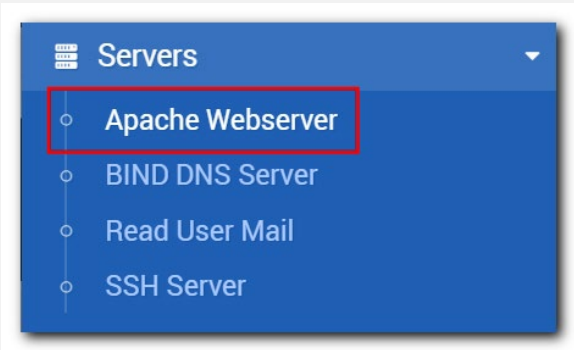




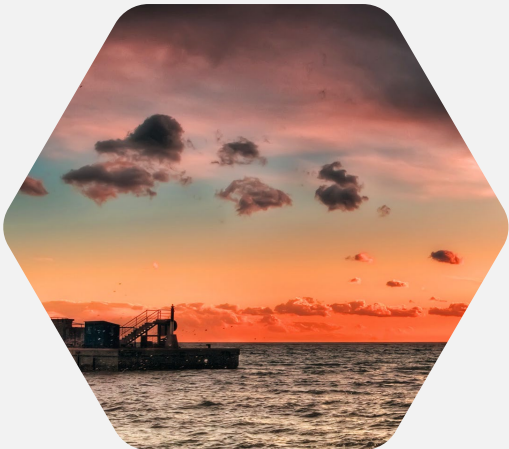
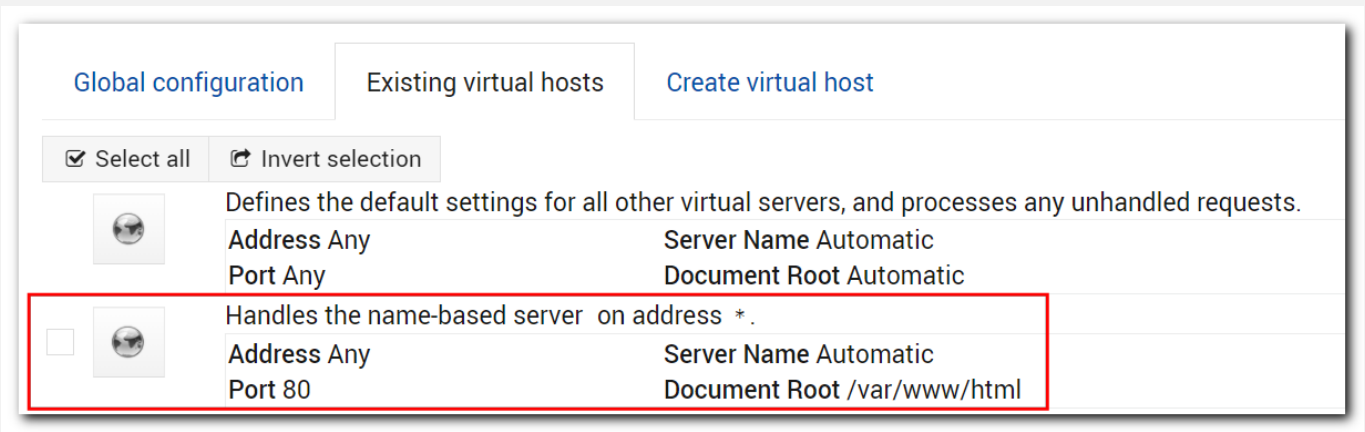
# AUTHENTICATION BASIC

## WEBMIN

Podemos realizar esta configuración desde **Webmin**



Lo haremos desde alguno de los servidores virtuales



# AUTHENTICATION BASIC

## WEBMIN

Elegimos o creamos un directorio a proteger

Per-Directory Options

Directory  
/var/www/html

Directory /var/www/html/protegido

Create Per-Directory, Files or Location Options

Type: Directory ▼

Regexp? ☒ Exact match ☐ Match regexp

Path:

+ Create

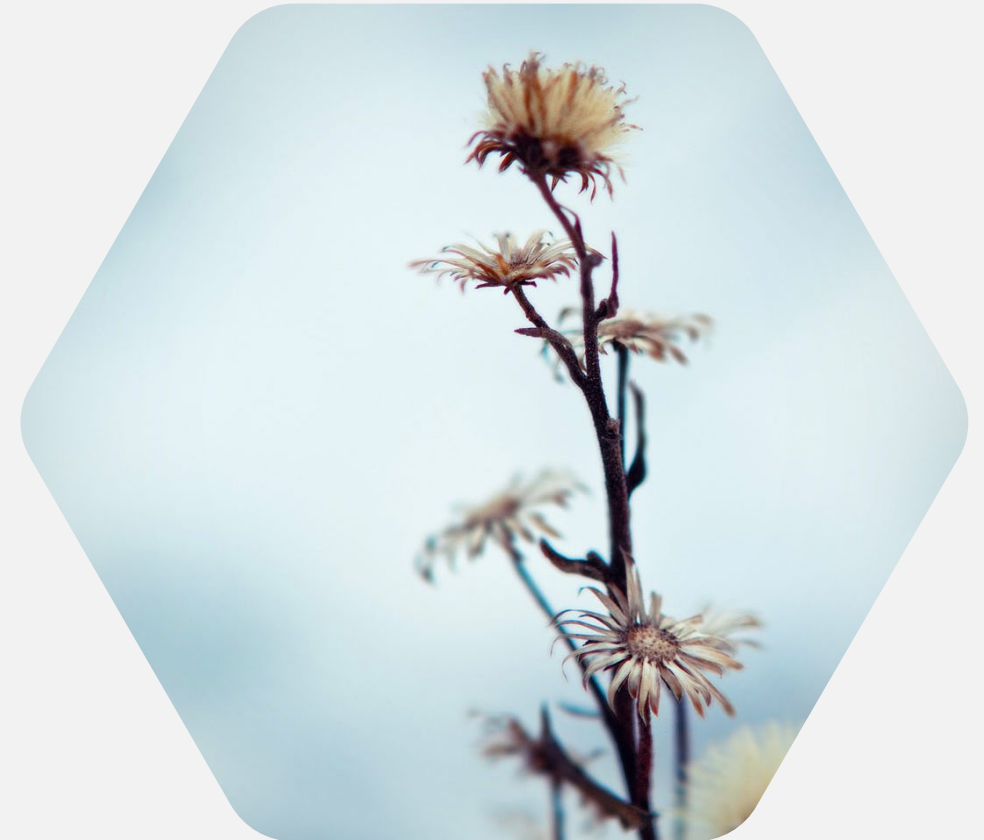
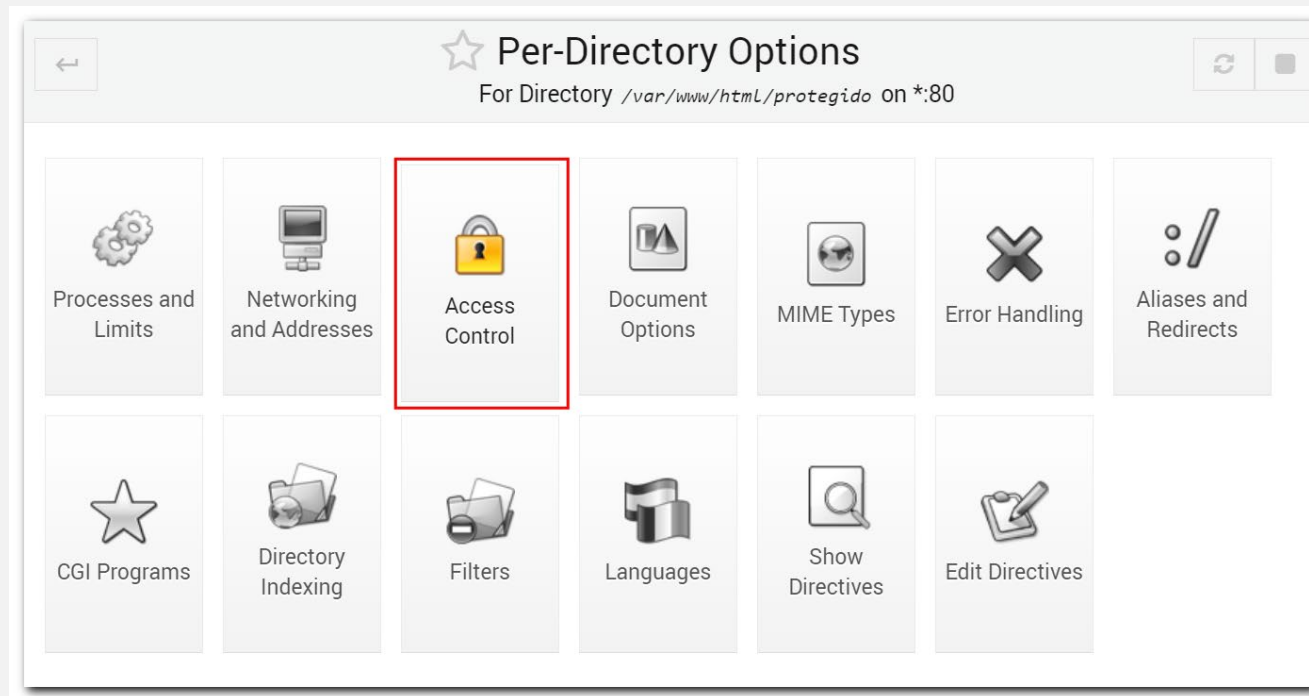




# AUTHENTICATION BASIC

## WEBMIN

Seleccionamos la opción **Access Control**





# AUTHENTICATION BASIC

## WEBMIN

Seleccionamos el conjunto de opciones de la figura

Access Control for Directory `/var/www/html/protegido`

Authentication realm name ☐ Default ☒ Area privada con Autenticatio

Restrict access by login ☐ Default  
☐ Only these users:   
☐ Only these groups:   
☒ All valid users

Pass basic login failures to next module? ☐ Yes ☐ No ☒ Default

Authentication type Basic ▼

Clients must satisfy ☒ Default  
☐ All access controls  
☐ Any access control

Basic login user file types  
Text file  
DBM database

User text file ☐ Default ☒ `/usr/local/apache/passwd/password`

[Edit users](#)

Restrict access Access checking order: ☐ Deny then allow ☐ Allow then deny ☐ Mutual failure ☒ Default

Action	Condition
▼	All requests ▼

# AUTHENTICATION BASIC

## WEBMIN

Es importante tener en cuenta que habrá que desmarcar en varias partes la opción "**Default**" para que la configuración se lleve a cabo



# AUTHENTICATION BASIC

## GRUPOS DE USUARIOS

Una vez creado el fichero de usuarios y contraseñas, podemos crear un **fichero de grupos de usuarios**

Para esto no se requiere ningún programa especial, ya que es un fichero en texto plano

El formato del fichero es el siguiente:

nombreGrupo1: usuario1 usuario2

nombreGrupo2: usuario3 usuario4 usuario5

Cada grupo puede tener un número de usuarios, que se separan por espacios

Para que funcione, los usuarios deben estar creados con **htpasswd**





# AUTHENTICATION BASIC

## GRUPOS DE USUARIOS

Podemos ubicar el fichero de grupos en la misma carpeta que el de contraseñas

Para usarlo, hay que incluir las directivas siguientes:

```
AuthGroupFile /ruta/al/fichero/de/grupos  
require group grupo1 grupo2
```

**NOTA:** Para que funcione, hay que habilitar el módulo de Apache **authz\_groupfile**



# AUTHENTICATION BASIC

## WEBMIN

```
<Directory "/var/www/protegido">  
    AuthType Basic  
    AuthBasicProvider file  
    AuthUserFile /usr/local/apache/passwd/passwords  
    AuthGroupFile /usr/local/apache/passwd/groups  
    AuthName "Area Privada"  
    Require group grupo1  
</Directory>
```

