

# Configuración de aplicaciones en Linux

## Sumario

- 1 Configuración de aplicaciones
  - ◆ 1.1 Manual vi
  - ◆ 1.2 Servidor MySQL
  - ◆ 1.3 Apache + PHP

## Configuración de aplicaciones

### Manual vi

- [Manual Interesante vi](#)

### Servidor MySQL

- Facer unha copia de seguridade dunha base de datos MySQL

```
mysqldump -u usuario -pxxxxxx -C -Q -e -a nomebasedatos > backup.sql
```

```
-- xxxxxx (contrasinal do usuario)
```

- Recuperar unha copia de MySQL

```
mysql -u usuario -pxxxxxx -pxxxxxx < backup.sql
```

```
-- xxxxxx (contrasinal do usuario)
```

```
-- Asegurarse que dentro do arquivo backup.sql temos a instrucción:  
use nomebasedatos;
```

```
-- Outro xeito de facelo mesmo podería ser:
```

```
mysql -u usuario -pxxxxxx
```

```
mysql>source backup.sql;
```

### Apache + PHP

- *Desactivar PHP Notice E PHP Warning*

En PHP, mediante o parámetro de configuración **error\_reporting** podemos definir que tipo de erros queremos almacenar no *log*.

No caso que o teñamos a **E\_ALL**, veremos como se van gardando no log **PHP Notice**, os cales moitas veces coñecemos peron non nos interesan nun servidor en produción. Podemos desactivalos "restando" de **E\_ALL** os notices mediante **E\_NOTICE**:

```
error_reporting = E_ALL & ~E_NOTICE
```

Outro tipo de erros que poden encher os logs son os **PHP Warning**. Para eliminalos podemos facelo "restando" á expresión anterior **E\_WARNING**:

```
error_reporting = E_ALL & ~E_NOTICE & ~E_WARNING
```

- **Activar el mod\_rewrite en Centos**

Para que Apache poida aplicar as directivas do arquivo .htaccess do sitio web por defecto /var/www/html debemos facer o seguinte:

```
# Editamos o arquivo httpd.conf  
$ vi /etc/httpd/conf/httpd.conf
```

```
# Accedemos á sección:
```

```
# <Directory "/var/www/html">
```

```
# Options Indexes FollowSymLinks
```

```
# cambiamos:
```

```
# AllowOverride None
```

```
# a
```

```
# AllowOverride All

# Order allow,deny
# Allow from all
# </Directory>

# Logo sálvanse os cambios
$ service httpd restart
```

## • .htaccess

A definición de '.htaccess que podemos atopar en Wikipedia é a seguinte:

".htaccess (Acceso de Hiper-Texto) é o nome por defecto do arquivo de configuración de directorios de Apache. Nos permite personalizar a configuración das directivas definidas no arquivo de configuración principal. As directivas de configuración precisan estar no contexto de .htaccess e o usuario precisa os permisos axeitados."

### - Redireccionar a páxina de mantemento con .htaccess:

Cando estamos realizando cambios críticos na nosa web, vémonos na necesidade de redireccionar aos nosos visitantes a unha páxina de mantemento.

Isto non é maior problema, pero neste caso imos diferenciar a nosa ip: A todos os visitantes que non teñan a IP que indiquemos (a do noso equipo), redireccionaralles á páxina de mantemento mentres que nos veremos a páxina correctamente.

O arquivo .htaccess será o seguinte:

```
Options +FollowSymlinks
RewriteEngine On
RewriteCond %{REQUEST_URI} !/mantenimiento.html$
RewriteCond %{REMOTE_HOST} !^10\.10\.10\.10
RewriteRule $ /mantenimiento.html [R=302,L]
```

### - Control de acceso a carpetas:

Se queres deshabilitar totalmente o acceso a unha carpeta (por exemplo, unha carpeta con librerías de programación que se inclúen nos arquivos principais. Neste caso só os arquivos principais accederán a eles mediante o sistema de arquivos, pero non se poderá acceder a eles vía web). Para conseguilo só hai que crear un arquivo .htaccess nesa carpeta que conteña o seguinte:

```
#deny all access
deny from all
```

Se se quere permitir o acceso dende unha IP específica:

```
#deny all access
deny from all
allow from 10.0.0.1
```

Ou para un rango específico de IPs:

```
allow from 192.168.0.0/24
```

Tamén se pode bloquear o acceso a un arquivo específico:

```
<Files privado.html>
Order allow,deny
Deny from all
```

### - Listado de carpetas:

Se se quere facer que as carpetas sexan navegables, engadimos a seguinte liña:

```
Options +Indexes +MultiViews +FollowSymlinks
```

Se queremos previr o listado de carpetas:

```
IndexIgnore *
```

## - Carpetas Privadas con .htaccess y .htpasswd:

Se queremos protexer cun contrasinal certas carpetas dos sitios aloxados dentro do *Document Root* de Apache faremos os seguintes pasos:

1.- Editar o arquivo de configuración de Apache, normalmente **httpd.conf** e buscar na sección **Main server configuration** entre as etiquetas `<Directory />` a entrada **AllowOverride** e asignarlle o valor **AuthConfig**:

```
<Directory />
Options ExecCGI
AllowOverride AuthConfig
</Directory>
```

2.- Crear un arquivo **.htaccess** e escribir o seguinte:

```
AuthUserFile /etc/httpd/conf/.htpasswd //ruta absoluta do arquivo .htpasswd
AuthGroupFile /dev/null
AuthName "Nome do dominio"
AuthType Basic
<Limit GET POST>
require valid-user
</Limit>
```

O arquivo **.htaccess** gardarase dentro das carpetas a protexer con contrasinal.

3.- Crear o arquivo **.htpasswd**: Para isto hai que executar o comando **htpasswd** que ven con Apache e que encripta o contrasinal cunha versión modificada do algoritmo MD5 e empregando a función **crypt** no caso de Linux polo que non podemos crealas directamente. Olo, estes arquivos non serán compatibles entre Windows e Linux.

O comando a executar para o arquivo é:

```
htpasswd -c .htpasswd pepe
```

Para engadir un usuario:

```
htpasswd .htpasswd xan
```

**O programa nos pedirá o password automáticamente.**

Se queremos que o programa nos mostre a entrada por pantalla para posibles *scripts* debemos empregar o comando:

```
htpasswd -nb nome_usuario password
```

- [.htaccess en apache.org](http://.htaccess.en.apache.org)