

# LAB08

## Protocolo HTTP Servidor Apache

---

### A. Protocolo HTTP

Neste exercício vamos utilizar a aplicação `telnet`, ligar-nos directamente à porta TCP 80 do servidor web, e familiarizarmo-nos com os comandos do protocolo HTTP.

1. Corre a aplicação `telnet` e pede a página de entrada do servidor web

```
serverXX1:~#apt-get install telnet
serverXX:~#telnet 10.10.23.13 80
GET / HTTP/1.0
(linha em branco)
(linha em branco)
```

Que é que recebeste como resposta?

Linha de status: \_\_\_\_\_

Cabeçalhos:

Connection: \_\_\_\_\_

Content-Type: \_\_\_\_\_

Dados (html, pdf, jpg, png, ...): \_\_\_\_\_

```
serverXX:~#telnet 10.10.23.13 80
GET /images/planetlab1.gif HTTP/1.0
(linha em branco)
(linha em branco)
```

Que é que recebeste como resposta?

Linha de status: \_\_\_\_\_

Cabeçalhos:

Connection: \_\_\_\_\_

Content-Type: \_\_\_\_\_

Dados (html, pdf, jpg, png, ...): \_\_\_\_\_

2. Neste exercício vamos enviar dados para o servidor juntamente com o URL utilizando o comando `GET`.

`test2.pl` é um programa em Perl que se encontra no servidor e que devolve uma página html com todas as variáveis fornecidas pelo servidor web ao programa (ver o código em apêndice). É através da variável \_\_\_\_\_ que o programa tem conhecimento dos dados que nós enviamos.

---

<sup>1</sup> “serverXX” é o nome do teu servidor

Numero:

Nome:

Data:

```
serverXX:~#telnet 10.10.23.13 80
GET /cgi-bin/test2.pl?nome=bruno&idade=23 HTTP/1.0
(linha em branco)
(linha em branco)
```

Que é que recebeste como resposta?

QUERY\_STRING=\_\_\_\_\_

3. Neste exercício vamos enviar dados para o servidor web através do comando POST. `post.pl` é um programa em Perl que se encontra no servidor e que devolve uma página html com o corpo da mensagem HTTP (ver o código em apêndice)

```
serverXX:~#telnet 10.10.23.13 80
POST /cgi-bin/post.pl HTTP/1.0
Content-Length:22
(linha em branco)
ano=2003&mes=11&dia=20
(linha em branco)
```

Que é que recebeste como resposta?

Linha de status: \_\_\_\_\_

Cabeçalhos:

Connection: \_\_\_\_\_

Content-Type: \_\_\_\_\_

Dados (html, pdf, jpg, png, ...): \_\_\_\_\_

## B. Instalação do servidor Apache

4. Para instalar o servidor Apache basta executar o comando

```
serverXX:~# apt-get install apache2
```

Verifica que o serviço está a funcionar

```
serverXX:~# netstat -anp | grep tcp
serverXX:~# telnet 127.0.0.1 80
GET / HTTP/1.0
(linha em branco)
(linha em branco)
```

5. Verifica que o serviço está a funcionar com o browser (chromium-browser...) abrindo o URL `http://localhost/`.

Funciona? \_\_\_\_\_

- Vê o que está definido no ficheiro de configuração `/etc/apache2/apache2.conf`
- Vê quais os modulos que estão activos em `/etc/apache2/mods-enabled`
- Vê quais os sites que estão activos em `/etc/apache2/sites-enabled`

6. Altera a página de entrada do site para uma da tua preferência substituindo `/var/www/index.html` pela tua página html preferida.

**Por exemplo** guarda a página de entrada do Google (`www.google.pt`) utilizando o menu

```
(chromium-browser) Settings > Save Page as ... >
/var/www/google.html
```

Copia a página que acabaste de guardar para `/var/www/index.html`:

```
#mv /var/www/index.html /var/www/index.html.org
#mv /var/www/google.html /var/www/index.html
```

7. Faz refresh à página `http://localhost/`

Quais são as diferenças relativamente à página original do google? \_\_\_\_\_

O que deves fazer para aparecer o logotipo do google? \_\_\_\_\_

Qual o nome do serviço e o IP do servidor que tens que “assaltar” (hijack) para re- direccionares os acessos ao google para o teu servidor?

Nome do serviço: \_\_\_\_\_

IP do servidor: \_\_\_\_\_. \_\_\_\_\_. \_\_\_\_\_. \_\_\_\_\_

### C. Criação do directório web pessoal de um utilizador

8. Activa o modulo "userdir" do apache

```
# cd /etc/apache2/mods-enabled
/etc/apache2/mods-enabled# ln -s ../mods-available/userdir.conf userdir.conf
/etc/apache2/mods-enabled# ln -s ../mods-available/userdir.load userdir.load
```

Faz restart ao serviço

```
serverXX:~# /etc/init.d/apache2 restart
```

9. Cria o utilizador asterix

```
serverXX:~# adduser asterix
serverXX:~# chmod a+x /home/asterix
```

10. Muda para o utilizador asterix e cria o directório public\_html

```
serverXX:~# su - asterix
asterix@serverXX:~$ mkdir public_html
asterix@serverXX:~$ chmod a+rx public_html
```

11. Copia a tua página web preferida para dentro do directório public\_html

```
asterix@serverXX:~$ cp /var/www/index.html ~/public_html
```

12. Verifica com o browser que a página web pessoal do utilizador asterix está agora acessível em <http://localhost/~asterix>

Sucesso? \_\_\_\_\_

13. Qual a directiva em `userdir.conf` que terás que mudar para que a pasta web pessoal `public_html` passe a chamar-se `web_folder` ?

directiva: \_\_\_\_\_

### D. Criação de um directório com acesso restrito

14. Cria a pasta "privado" do utilizador asterix

```
asterix@serverXX:~$ mkdir ~/public_html/privado
asterix@serverXX:~$ chmod a+x ~/public_html/privado
```

15. Cria o ficheiro `~/public_html/privado/.htaccess` com o seguinte conteúdo

```
asterix@serverXX:~$ nano ~/public_html/privado/.htaccess
```

Numero:

Nome:

Data:

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

16. Cria o ficheiro .htpasswd

```
asterix@serverXX:~$ cd ~/public_html/privado/
$ /usr/bin/htpasswd -c -b .htpasswd asterix segredo
```

17. Verifica que te é pedida uma password para aceder ao URL

<http://localhost/~asterix/privado>

Qual a directiva que debes incluir no ficheiro `.htaccess` para garantir que o asterix é um utilizador válido do servidor? \_\_\_\_\_

Quais são as directivas que estão dentro do contentor `<Files ~ "^\.ht" >` que impedem que se possa fazer download do ficheiro `.htpasswd` ?

\_\_\_\_\_

(Sugestão: a tua página web pessoal no DEEI <http://intranet.deei.fct.ualg.pt/~axxxxx/>, pode ser protegida por password utilizando os comando que aprendeste neste guião ...)

## E. Criação de dois hosts virtuais

18. No servidor de DNS\* 10.11.160.1 existem dois “aliases” adicionais para cada servidor na sala

```
girlsXX  IN  CNAME  serverXX.grs.deei.
boysXX   IN  CNAME  serverXX.grs.deei.
```

**Assegura-te** que o servidor de DNS 10.11.160.1 é o *primeiro* a ser consultado:

```
serverXX:~# nano /etc/resolv.conf
search grs.deei
nameserver 10.11.160.1
```

Verifica!

```
serverXX:~# nslookup -type=CNAME boysXX.grs.deei
serverXX:~# nslookup -type=CNAME girlsXX.grs.deei
```

19. Cria as pastas girls e boys em /var/www

```
serverXX:~# mkdir /var/www/girls
serverXX:~# mkdir /var/www/boys
serverXX:~# chmod a+rx /var/www/girls
serverXX:~# chmod a+rx /var/www/boys
```

20. Cria os ficheiros “boys.grs.deei” e “girls.grs.deei”

```
serverXX:~# nano /etc/apache2/sites-available/boys.grs.deei
```

```
<VirtualHost *:80>
    ServerName      boysXX.grs.deei
    DocumentRoot    /var/www/boys/
</VirtualHost>
```

- 
- Nota: Se o servidor de DNS não funcionar/não existir, no mínimo é preciso que existam entradas para estes nomes no ficheiro /etc/hosts!

```
server__:~# nano /etc/hosts
127.0.0.1      localhost
127.0.0.1      serverXX.grs.deei  serverXX
127.0.0.1      girlsXX.grs.deei  girlsXX
127.0.0.1      boysXX.grs.deei   boysXX
```

Numero:

Nome:

Data:

```
serverXX:~# nano /etc/apache2/sites-available/girls.grs.deei
```

```
<VirtualHost *:80>
    ServerName      girlsXX.grs.deei
    DocumentRoot    /var/www/girls/
</VirtualHost>
```

E activa os “virtual hosts” :

```
serverXX:~# cd /etc/apache2/sites-enabled
/etc/apache2/sites-enabled# ln -s ../sites-available/boys.grs.deei boys.grs.deei
/etc/apache2/sites-enabled# ln -s ../sites-available/girls.grs.deei girls.grs.deei
```

Re-inicia o apache

```
serverXX:~# /etc/init.d/apache2 restart
```

21. Coloca dois ficheiros index.html **diferentes** em /var/www/girls e /var/www/boys.

22. Verifica com o browser que os URLs

```
http://serverXX.grs.deei/
http://boysXX.grs.deei/
http://girlsXX.grs.deei/
```

**devolvem páginas diferentes**, apesar dos três sites estarem alojadas no mesmo servidor web...

Sucesso?\_\_\_\_\_

Faz o debugging necessario até que os dois “virtual hosts” estejam a funcionar correctamente ...

23. Termina aqui este laboratório.

## APÊNDICE

### test2.pl

```
#!/usr/bin/perl

print "Content-type:text/html\n\n";
print <<EndOfHTML;
<html><head><title>Print Environment</title></head>
<body>
EndOfHTML

foreach $key (sort(keys %ENV)) {
    print "$key = $ENV{$key}<br>\n";
}

print "</body></html>";
```

### post.pl

```
#!/usr/bin/perl

print "Content-type:text/html\n\n";

read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs) {
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+//;
    $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
    $FORM{$name} = $value;
}

print "<html><head><title>Form Output</title></head><body>";
print "<h2>Results from FORM post</h2>\n";

foreach $key (keys(%FORM)) {
    print "$key = $FORM{$key}<br>";
}

print "</body></html>";
```