



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Administración de servicios en red



Misión 2: Servidor HTTP

Documento de pruebas

Equipo 9:

Bryan Dominguez de la Rosa

Rocío Vivia Delgadillo

Fernando Jair Pacheco Díaz

Grupo: 4CV3

Servidor HTTP

Para las pruebas descritas a continuación se utilizó un servidor HTTP Apache versión 2.4, montado sobre Ubuntu 18.04 LTS de 64 bits en un equipo con procesador Intel Core i3 a 2.20Ghz, con 4GB de Ram. Se montaron 3 contenedores, cada uno con un sitio web diferente, solo se realizaron las pruebas en 2 de ellos.

CONTENEDOR 1

En este contenedor se encuentra el sitio www.example.com, el cuál consta de una página sencilla con JavaScript para unas validaciones de diseño.

ACCESO A CONTENEDOR VIRTUAL POR DOMINIO

Al ingresar la url en un navegador en un equipo externo, conectado a la misma red local, éste encuentra el recurso adecuado.



LOG DE ACCESO A CONTENEDOR VIRTUAL POR DOMINIO

Y en el archivo `.log` correspondiente a ese contenedor, observamos que en efecto, recibió la petición GET y devolvió el recurso solicitado, acompañado de un código 200.

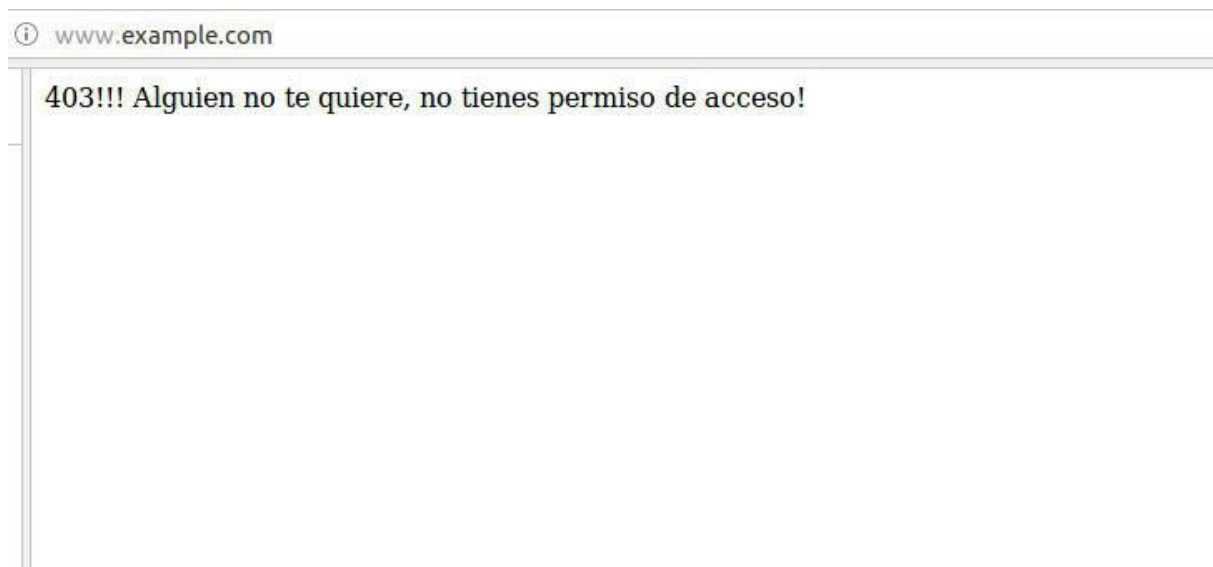
```
jair@lap-jair:/var/log/apache2$ tail -f example.log
10.100.67.208 - - [20/Sep/2018:18:17:40 -0500] "GET / HTTP/1.1" 200 943 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0"
10.100.67.208 - - [20/Sep/2018:18:17:40 -0500] "GET /p2.css HTTP/1.1" 200 1053 "http://www.example.com/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0"
```

RESTRICCIÓN POR IP DEL CLIENTE

Al modificar la configuración del servidor, en el archivo `apache2.conf`, ingresando las siguientes líneas, impedimos que este pueda acceder a un recurso en específico o al sitio entero.

```
<Directory /var/www/example.com>
  Options all
  AllowOverride all
  <RequireAll>
    Require all granted
    Require not ip 10.100.70.182
  </RequireAll>
</Directory>
```

El resultado en pantalla es el siguiente error 403:



LOG DE RESTRICCIÓN POR IP

Se observa que el servidor respondió a esa IP restringida con un código 403.

```
10.100.67.208 - - [20/Sep/2018:18:25:25 -0500] "GET / HTTP/1.1" 403 423 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0"
```

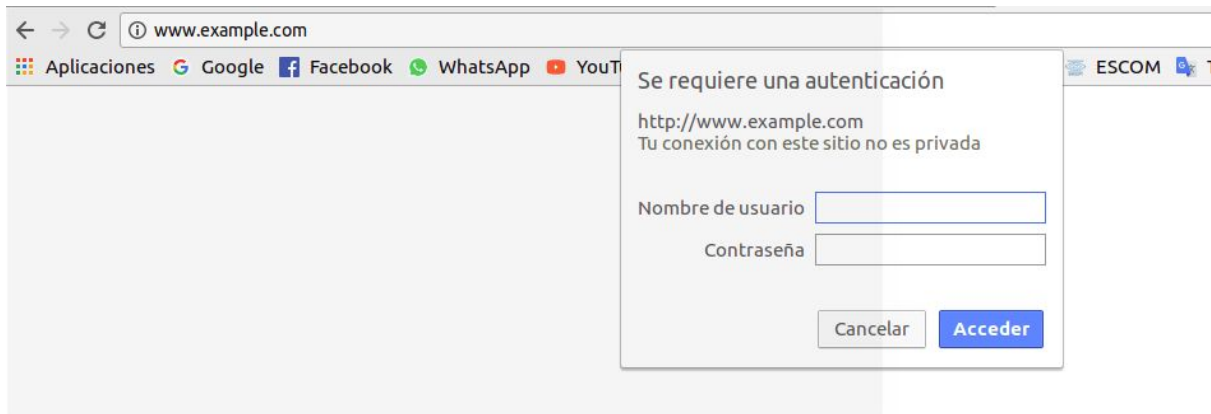
RESTRINGIR POR SEGMENTO

En el mismo documento de configuración, se redactan las siguientes líneas y ninguna IP del 10.100.70.0 al 10.100.70.255 podrá acceder al sitio, mostrando el mismo error 403.

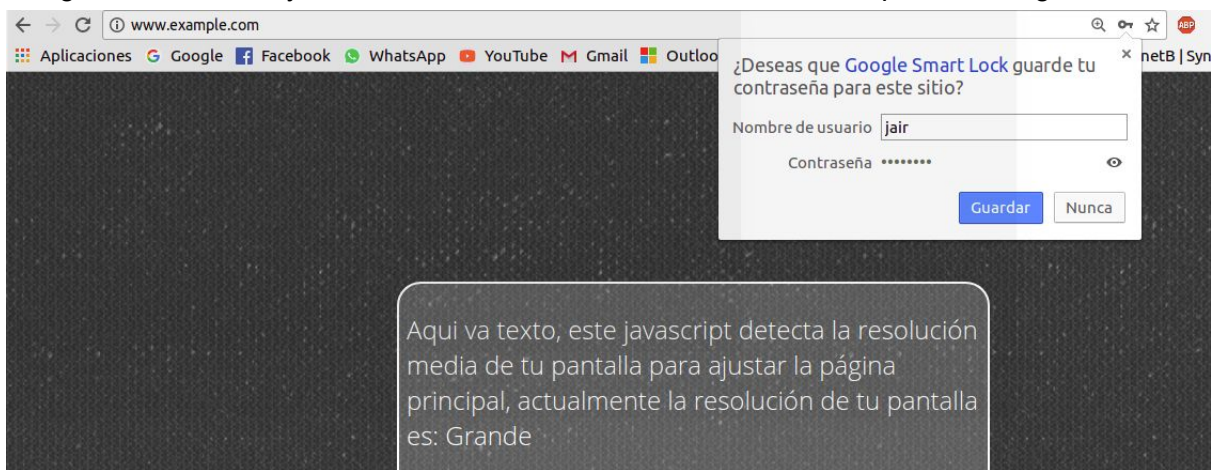
```
<Directory /var/www/example.com>
  Options all
  AllowOverride all
  <RequireAll>
    Require all granted
    Require not ip 10.100.70
  </RequireAll>
</Directory>
```

RESTRINGIR POR NOMBRE DE USUARIO

Para que el sitio solicite un usuario y una contraseña antes de acceder a este, se modifica el mismo archivo de configuración, añadiendo un archivo creado externamente que contenga el usuario y la contraseña (Con htpasswd se genera encriptada).



Al ingresar el usuario y la contraseña correctos, se entra al sitio sin problema alguno.



CONFIGURACIÓN RESTRICCIÓN POR NOMBRE DE USUARIO

Se debe tener acceso al archivo en el que se encuentre los datos del usuario.

```
<Directory /var/www/example.com>
  Options all
  AllowOverride all
  <RequireAll>
    AuthName "Private"
    AuthType Basic
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
  </RequireAll>
</Directory>
```

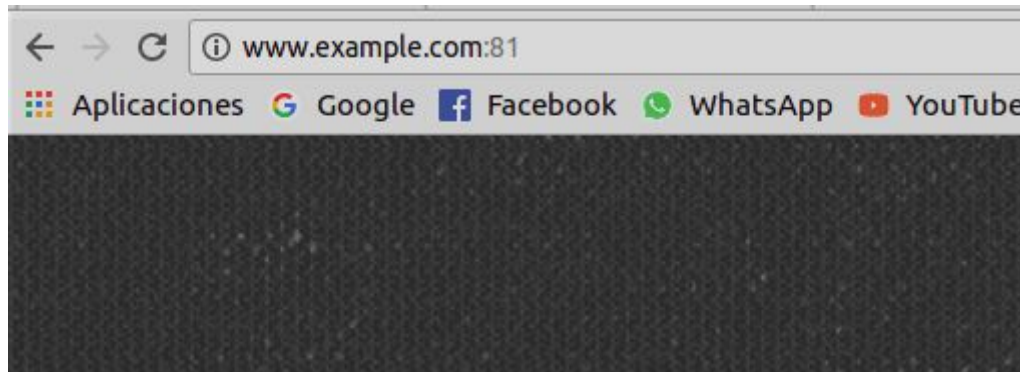

LOG RESTRICCIÓN POR USUARIO

El log correspondiente muestra la respuesta primero un 401 para solicitar los datos de autenticación y al hacerlo correctamente regresa un código 200.

```
10.100.70.182 - jair [20/Sep/2018:18:41:07 -0500] "GET / HTTP/1.1" 401 445 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36"
10.100.70.182 - jair [20/Sep/2018:18:41:17 -0500] "GET / HTTP/1.1" 200 943 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36"
```

ACCESO A PUERTO CONFIGURADO

También se configuró el sitio para que respondiera al puerto distinto al 80 que es el puerto por defecto de las peticiones HTTP, de tal manera que se acceda así al sitio:



CONFIGURACIÓN PUERTO DE OPERACIÓN

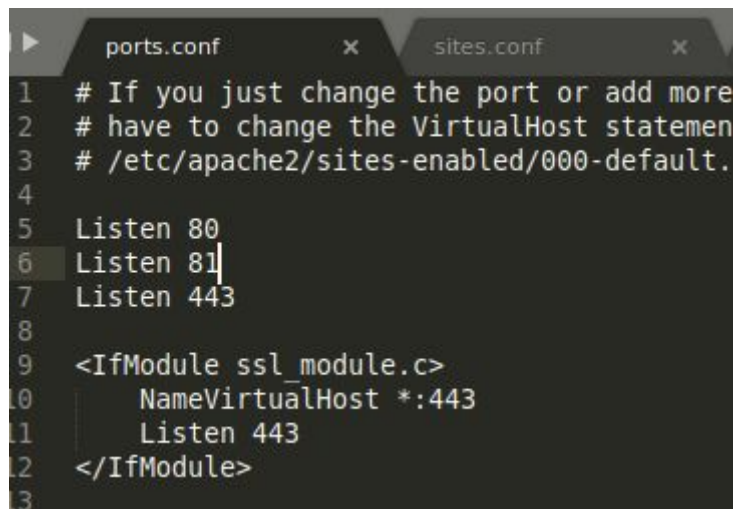
Para esto vamos al archivo de configuración del host virtual donde está el sitio, al inicio de este, se cambia el *:80 por *:81.

```
<VirtualHost *:81>
    ServerAdmin admin@sites.com
    ServerAlias example.com
    ServerName www.example.com
    DocumentRoot /var/www/example.com/html

    ErrorLog ${APACHE_LOG_DIR}/errorExample.log
    CustomLog ${APACHE_LOG_DIR}/example.log combined

    ErrorDocument 401 /error_401.html
    ErrorDocument 403 /error_403.html
    ErrorDocument 404 /error_404.html
</VirtualHost>
```

Posteriormente, se va al archivo de ports.conf de Apache y con el comando Listen, se pone que también escuche dicho puerto para responder peticiones, de la siguiente manera:



```

1 # If you just change the port or add more
2 # have to change the VirtualHost statemen
3 # /etc/apache2/sites-enabled/000-default.
4
5 Listen 80
6 Listen 81
7 Listen 443
8
9 <IfModule ssl_module.c>
10     NameVirtualHost *:443
11     Listen 443
12 </IfModule>
13

```

CONFIGURACIÓN DEL HTTPS

Se crearon nuevos virtualhost que apunten al mismo sitio, pero usando los archivos de certificación correspondientes, las peticiones HTTPS por defecto deben ir al puerto 443. Con los comandos de SSL se colocan las rutas de los certificados a usar.

```

<IfModule mod_ssl.c>
    <VirtualHost *:443>
        ServerAdmin admin@sites.com
        ServerAlias example.com
        ServerName www.example.com
        DocumentRoot /var/www/example.com/html

        ErrorLog ${APACHE_LOG_DIR}/errorExample.log
        CustomLog ${APACHE_LOG_DIR}/example.log combined

        SSLEngine on
        SSLCertificateFile /etc/apache2/ssl/example.com.crt
        SSLCertificateKeyFile /etc/apache2/ssl/example.com.key

        <FilesMatch "\.(cgi|shtml|phtml|php)$">
            SSLOptions +StdEnvVars
        </FilesMatch>

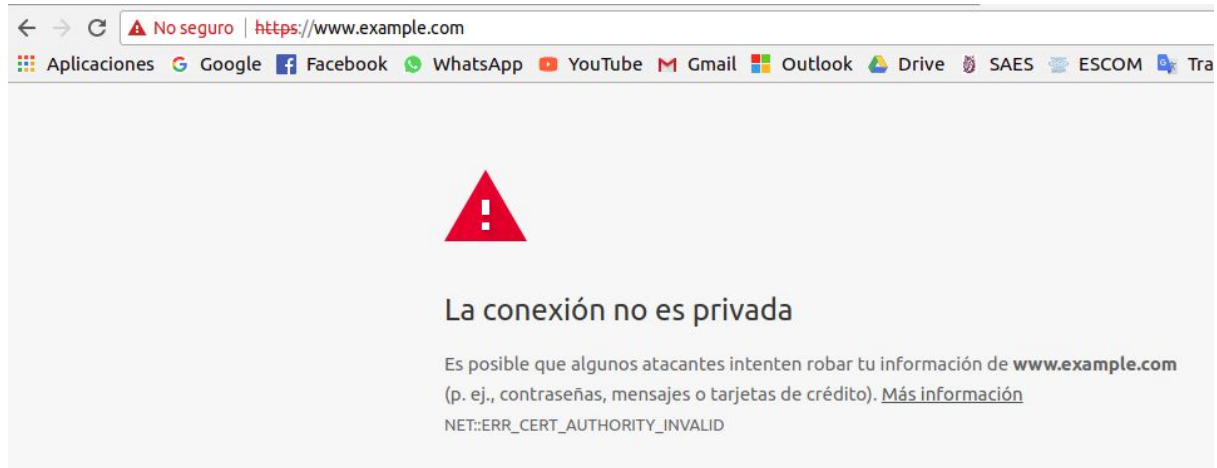
        <Directory /usr/lib/cgi-bin>
            SSLOptions +StdEnvVars
        </Directory>

    </VirtualHost>
</IfModule>

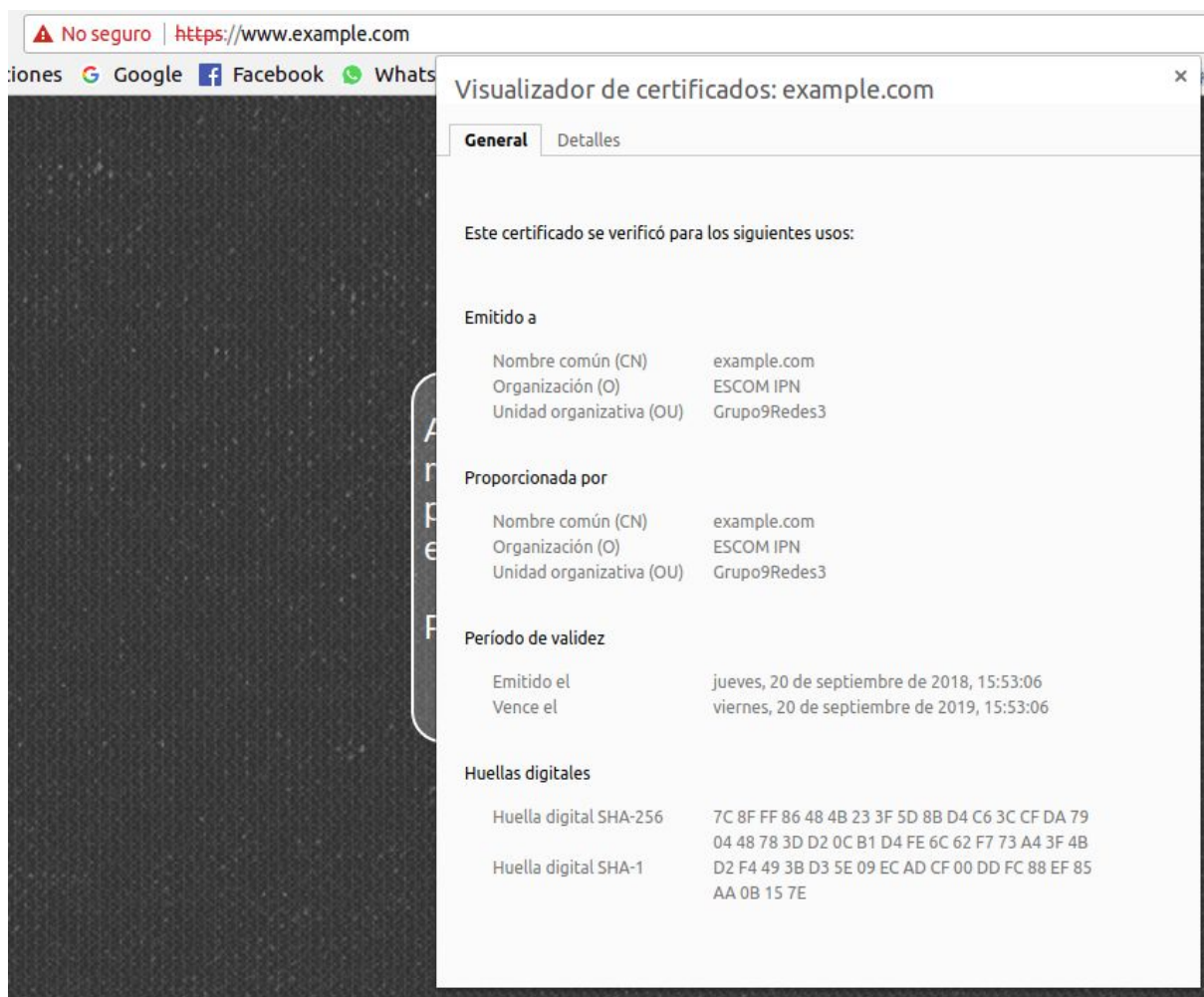
```

ACCESO CON HTTPS

Al reiniciar el server y entrar desde el equipo externo, al no ser un certificado válido, sale un mensaje de advertencia.



Como confiamos en nuestro certificado aún así, le damos continuar y podemos ver los detalles de dicho certificado.



TRAMA DE ELEMENTO CIFRADO

Utilizando software como Wireshark podemos observar las respuestas de las peticiones cifradas por la llave del certificado.

No.	Time	Source	Destination	Protocol	Length	Info
116	3.261931477	10.100.74.233	10.100.70.182	TLSv1.2	214	Server Hello, Change Cipher Spec, Encrypted Handshake Message
117	3.262126995	10.100.74.233	10.100.70.182	TLSv1.2	214	Server Hello, Change Cipher Spec, Encrypted Handshake Message
120	3.266311255	10.100.74.233	10.100.70.182	TLSv1.2	214	Server Hello, Change Cipher Spec, Encrypted Handshake Message
128	3.275736178	10.100.74.233	10.100.70.182	TLSv1.2	89	Encrypted Alert
132	3.276199558	10.100.74.233	10.100.70.182	TLSv1.2	89	Encrypted Alert
133	3.276299204	10.100.74.233	10.100.70.182	TLSv1.2	89	Encrypted Alert
147	3.288097248	10.100.74.233	10.100.70.182	TLSv1.2	214	Server Hello, Change Cipher Spec, Encrypted Handshake Message
152	3.290404416	10.100.74.233	10.100.70.182	TLSv1.2	536	Application Data
112	3.261467392	10.100.70.182	10.100.74.233	TLSv1.2	605	Client Hello
114	3.261505997	10.100.70.182	10.100.74.233	TLSv1.2	605	Client Hello
118	3.265994216	10.100.70.182	10.100.74.233	TLSv1.2	605	Client Hello
123	3.275567060	10.100.70.182	10.100.74.233	TLSv1.2	100	Change Cipher Spec, Encrypted Handshake Message
125	3.275588083	10.100.70.182	10.100.74.233	TLSv1.2	100	Change Cipher Spec, Encrypted Handshake Message

▶ Frame 152: 536 bytes on wire (4288 bits), 536 bytes captured (4288 bits) on interface 0
▶ Ethernet II, Src: LiteonTe_ae:2d:63 (30:52:cb:ae:2d:63), Dst: LiteonTe_4b:00:04 (ac:e0:10:4b:00:04)
▶ Internet Protocol Version 4, Src: 10.100.74.233, Dst: 10.100.70.182
▶ Transmission Control Protocol, Src Port: 443, Dst Port: 56736, Seq: 149, Ack: 1036, Len: 470
▶ Secure Sockets Layer





0000	ac e0 10 4b 00 04 30 52	cb ae 2d 63 08 00 45 00	...K..0R ...c..E.
0010	02 0a 77 bf 40 00 40 06	1a c8 0a 64 4a e9 0a 64	..w.0.0. ...dJ..d
0020	46 b6 01 bb dd a0 55 9f	0b 97 33 69 16 50 80 18	F.....U. ...3i.P..
0030	00 f4 ed ca 00 00 01 01	08 0a c5 56 55 50 04 edVUP..
0040	20 e4 17 03 03 01 d1 f0	e4 3e 45 d2 d9 d6 3d ce >E...=.
0050	13 b2 52 ae 8b df e8 3f	c1 73 f9 04 ae aa 3b 55	..R....? ..s....;U
0060	d0 eb 54 5f 74 26 62 17	b7 e2 7b 07 1f 58 d7 19	..T_t&b. ...{.X..
0070	cc 25 bc 98 f3 1f 23 0f	48 71 24 91 e6 bb 62 c7	%.#. Hq\$. ...b.
0080	00 80 2d 03 20 db 98 97	07 cb 2d 1b 33 a0 72 e03.r.
0090	10 a7 af 6a 16 e0 8e 4f	bd e3 91 9f 80 f7 09 74	...j...0t
00a0	83 cc b2 f6 de 00 9c 25	58 63 41 98 44 f1 00 c0% XcA.D...
00b0	48 5e 51 16 d3 2d 44 5a	dd a3 78 16 f4 ae fe fd	H^Q...-Z ..x....
00c0	40 18 25 d9 88 bb 0d 39	4d 17 96 dd 6c 7c e0 bb	@.%....9 M...l ..
00d0	f7 53 79 04 3c 73 56 28	d6 6e 05 95 08 ff fc ef	.Sy.<sV(.n.....

CONTENEDOR 2

Para el contenedor 2 se utilizó un sitio llamado www.secondexample.com, este sitio a pesar de no contar con características notables de diseño, hace uso de PHP y MySQL, ambos módulos correctamente instalados en el servidor.

ACCESO A CONTENEDOR VIRTUAL POR DOMINIO

[←](#) [→](#) [↻](#) [www.secondexample.com](#)

 Aplicaciones  Google  Facebook  WhatsApp

Form

Nombre:

Grupo:

LOG DE ACCESO A CONTENEDOR VIRTUAL POR DOMINIO

Se repite el proceso que se hizo en el primero contenedor y podemos acceder al sitio usando el dominio.

```
10.100.70.182 - - [20/Sep/2018:18:53:21 -0500] "GET / HTTP/1.1" 200 449 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36"
```

Para las restricciones no cambiaron los pasos, a continuación los resultados de manera gráfica son mostrados.

RESTRICCIÓN POR IP DEL CLIENTE

```
<Directory /var/www/secondexample.com>
  Options all
  AllowOverride all
  <RequireAll>
    Require all granted
    Require not ip 10.100.70.182
  </RequireAll>
</Directory>
```



LOG DE RESTRICCIÓN POR IP

```
10.100.70.182 - - [20/Sep/2018:18:55:42 -0500] "GET / HTTP/1.1" 403 423 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36"
```

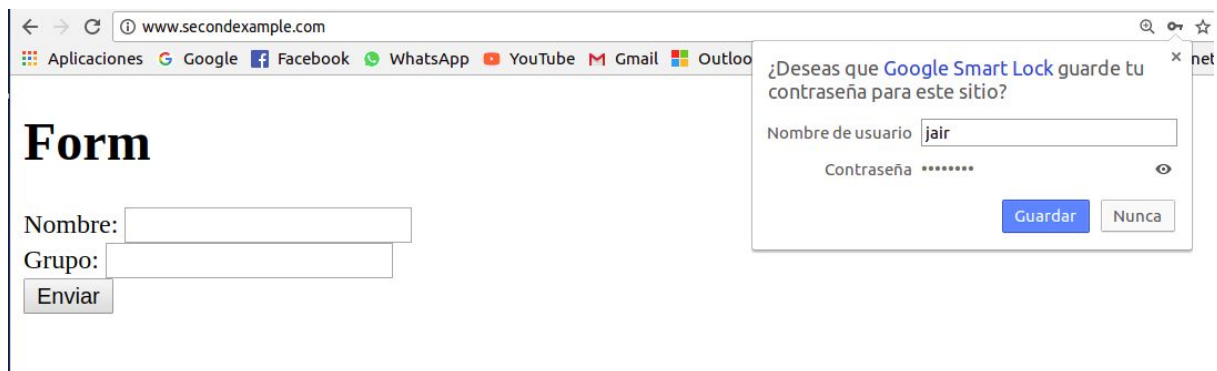
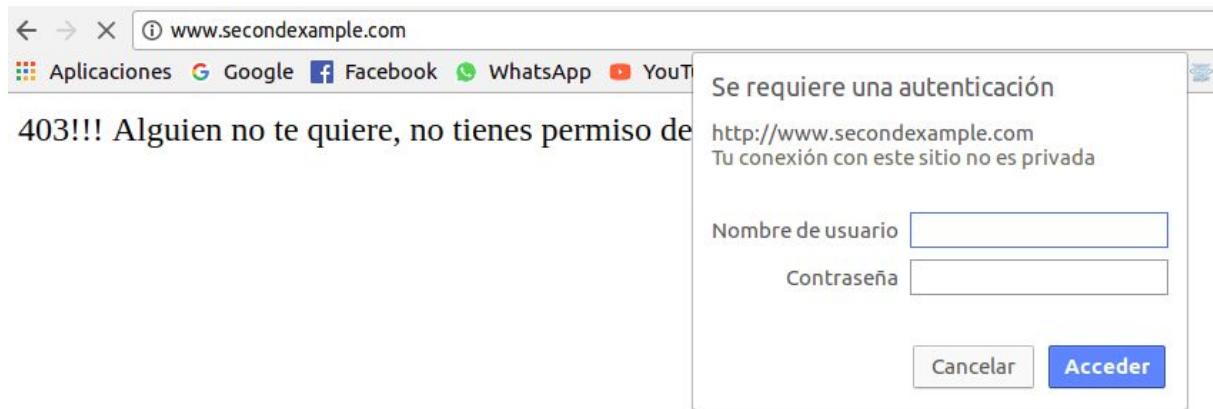
LOG RESTRICCIÓN POR SEGMENTO

```
10.100.70.182 - - [20/Sep/2018:18:57:38 -0500] "GET / HTTP/1.1" 403 423 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36"
```

RESTRICCIÓN POR SEGMENTO

```
<Directory /var/www/secondexample.com>
  Options all
  AllowOverride all
  <RequireAll>
    Require all granted
    Require not ip 10.100.70
  </RequireAll>
</Directory>
```

RESTRINGIR POR NOMBRE DE USUARIO



CONFIGURACIÓN RESTRICCIÓN POR NOMBRE DE USUARIO

```
185 <Directory /var/www/secondexample.com>
186     Options all
187     AllowOverride all
188     <RequireAll>
189         AuthName "Private"
190         AuthType Basic
191         AuthUserFile /etc/apache2/.htpasswd
192         Require valid-user
193     </RequireAll>
194 </Directory>
```

LOG RESTRICCIÓN POR NOMBRE DE USUARIO

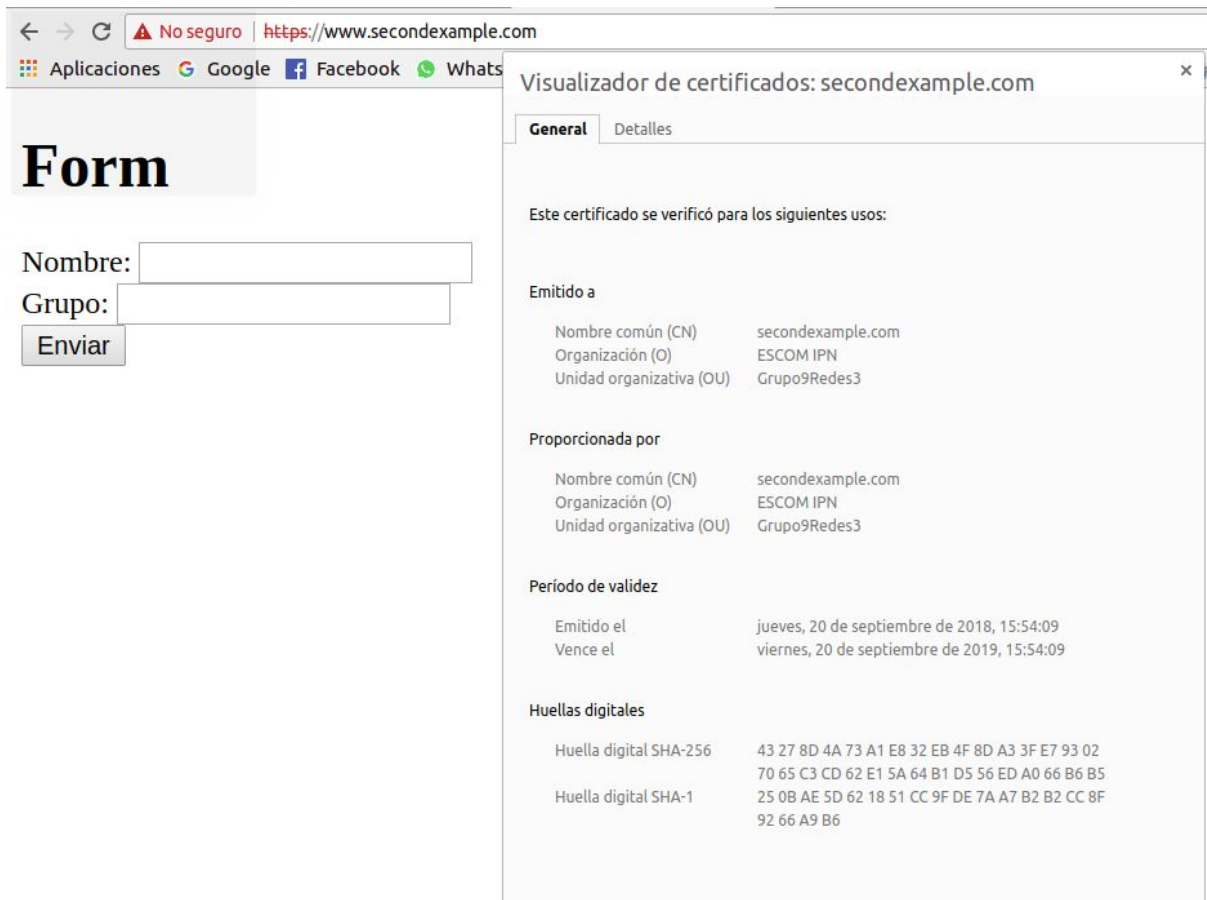
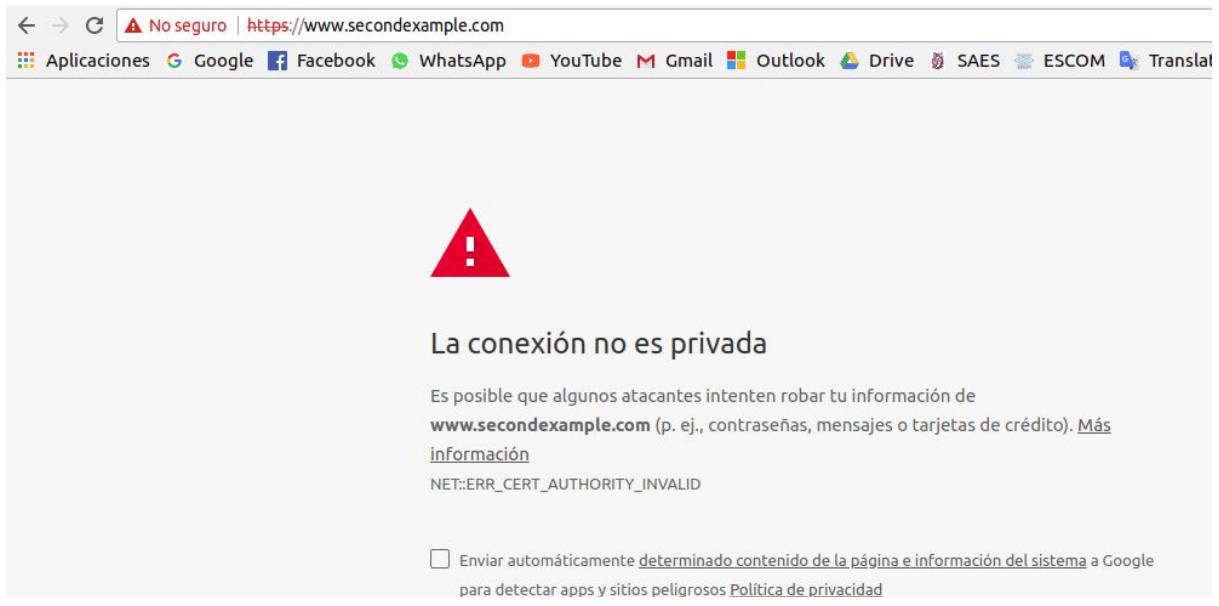
```
10.100.70.182 - - [20/Sep/2018:18:59:47 -0500] "GET / HTTP/1.1" 401 445 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36"
10.100.70.182 - jair [20/Sep/2018:19:01:13 -0500] "GET / HTTP/1.1" 200 449 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36"
```

CONFIGURACIÓN DEL PUERTO DE OPERACIÓN

El servidor anteriormente fue modificado para responder a las peticiones por el puerto 81, es el mismo proceso para ese contenedor.

ACCESO CON HTTPS

De la misma forma, se configuran los hosts virtuales para soportar el certificado de seguridad, pero como se observa a continuación, no se usaron los mismos archivos, si no que se generaron nuevos certificados de seguridad exclusivamente para este sitio.



TRAMA DE ELEMENTO CIFRADO

Y con Wireshark podemos observar los detalles de un paquete de la respuesta del servidor al cliente usando el certificado de seguridad.

No.	Time	Source	Destination	Protocol	Length	Info
116	3.261931477	10.100.74.233	10.100.70.182	TLSv1.2	214	Server Hello, Change Cipher Spec, Encrypted Handshake Message
117	3.262126995	10.100.74.233	10.100.70.182	TLSv1.2	214	Server Hello, Change Cipher Spec, Encrypted Handshake Message
120	3.266311255	10.100.74.233	10.100.70.182	TLSv1.2	214	Server Hello, Change Cipher Spec, Encrypted Handshake Message
128	3.275736178	10.100.74.233	10.100.70.182	TLSv1.2	89	Encrypted Alert
132	3.276199558	10.100.74.233	10.100.70.182	TLSv1.2	89	Encrypted Alert
133	3.276299204	10.100.74.233	10.100.70.182	TLSv1.2	89	Encrypted Alert
147	3.288097248	10.100.74.233	10.100.70.182	TLSv1.2	214	Server Hello, Change Cipher Spec, Encrypted Handshake Message
152	3.296404416	10.100.74.233	10.100.70.182	TLSv1.2	536	Application Data
112	3.261467392	10.100.70.182	10.100.74.233	TLSv1.2	605	Client Hello
114	3.261505997	10.100.70.182	10.100.74.233	TLSv1.2	605	Client Hello
118	3.265994216	10.100.70.182	10.100.74.233	TLSv1.2	605	Client Hello
123	3.275567060	10.100.70.182	10.100.74.233	TLSv1.2	100	Change Cipher Spec, Encrypted Handshake Message
125	3.275588083	10.100.70.182	10.100.74.233	TLSv1.2	100	Change Cipher Spec, Encrypted Handshake Message

▶ Frame 152: 536 bytes on wire (4288 bits), 536 bytes captured (4288 bits) on interface 0
▶ Ethernet II, Src: LiteonTe_ae:2d:63 (30:52:cb:ae:2d:63), Dst: LiteonTe_4b:00:04 (ac:e0:10:4b:00:04)
▶ Internet Protocol Version 4, Src: 10.100.74.233, Dst: 10.100.70.182
▶ Transmission Control Protocol, Src Port: 443, Dst Port: 56736, Seq: 149, Ack: 1036, Len: 470
▶ Secure Sockets Layer

0000	ac e0 10 4b 00 04 30 52	cb ae 2d 63 08 00 45 00	...K..0R ...c..E.
0010	02 0a 77 bf 40 00 40 06	1a c8 0a 64 4a e9 0a 64	..w.0.0. ...dJ..d
0020	46 b6 01 bb dd a0 55 9f	0b 97 33 69 16 50 80 18	F.....U. ...3i.P..
0030	00 f4 ed ca 00 00 01 01	08 0a c5 56 55 50 04 edVUP..
0040	20 e4 17 03 03 01 d1 f0	e4 3e 45 d2 d9 d6 3d ce>E...=.
0050	13 b2 52 ae 8b df e8 3f	c1 73 f9 04 ae aa 3b 55	..R....? .s....;U
0060	d0 eb 54 5f 74 26 62 17	b7 e2 7b 07 1f 58 d7 19	..T_t&b. ..{.X..
0070	cc 25 bc 98 f3 1f 23 0f	48 71 24 91 e6 bb 62 c7	%....#. Hq\$...b.
0080	00 80 2d 03 20 db 98 97	07 cb 2d 1b 33 a0 72 e0-..3.r.
0090	10 a7 af 6a 16 e0 8e 4f	bd e3 91 9f 80 f7 09 74	...j...0t
00a0	83 cc b2 f6 de 00 9c 25	58 63 41 98 44 f1 00 c0% XcA.D...
00b0	48 5e 51 16 d3 2d d4 5a	dd a3 78 16 f4 ae fe fd	H^Q...-Z .x.....
00c0	40 18 25 d9 88 bb 0d 39	4d 17 96 dd 6c 7c e0 bb	@%...9 M...l ..
00d0	f7 53 79 04 3c 73 56 28	d6 6e 05 95 08 ff fc ef	.Sy.<sV(.n.....

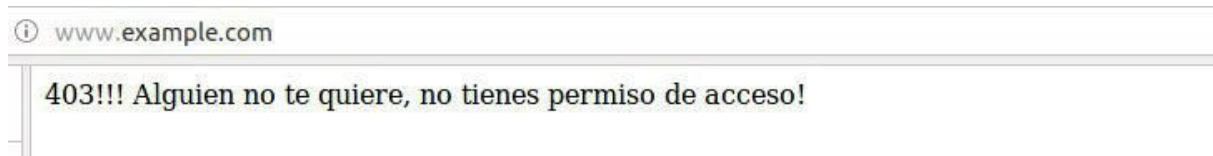
PERSONALIZACIÓN DE PÁGINAS DE ERROR

A continuación se presentan la personalización de las páginas que se devuelven al ocurrir un error, a lo largo del documento se han visto el uso de estas mismas.

ERROR 401



ERROR 403



ERROR 404



CONFIGURACIÓN DE ARCHIVOS DE ERRORES

Para configurarlos basta con colocar sus respectivas rutas en los hosts virtuales de los sitios.

```
12 <VirtualHost *:80>
13     ServerAdmin admin@sites.com
14     ServerAlias example.com
15     ServerName www.example.com
16     DocumentRoot /var/www/example.com/html
17
18     ErrorLog ${APACHE_LOG_DIR}/errorExample.log
19     CustomLog ${APACHE_LOG_DIR}/example.log combined
20
21     ErrorDocument 401 /error_401.html
22     ErrorDocument 403 /error_403.html
23     ErrorDocument 404 /error_404.html
24 </VirtualHost>
```

CONFIGURACIÓN DE ARCHIVOS DE BITÁCORAS

NIVEL DE CONFIGURACIÓN DE OPERACIÓN

Dentro del archivo `apache2.conf` es donde se colocan las banderas y el orden en el que los archivos logs son impresos, se pueden modificar de acuerdo a la documentación de Apache.

```
229 LogFormat "%v:%p %h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
230 LogFormat "%h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" combined
231 LogFormat "%h %l %u %t \"%r\" %>s %0" common
232 LogFormat "%{Referer}i -> %U" referer
233 LogFormat "%{User-agent}i" agent
```

CONFIGURACIÓN DE LOG DE ERROR POR SITIO

Cada host virtual tiene su archivo de log independiente, separando también el de errores y el de las peticiones y demás. A continuación las líneas utilizadas en los sitios probados en este documento.

```
ErrorLog ${APACHE_LOG_DIR}/errorExample.log
CustomLog ${APACHE_LOG_DIR}/example.log combined
```

```
31 ErrorLog ${APACHE_LOG_DIR}/errorSecondexample.log
32 CustomLog ${APACHE_LOG_DIR}/secondexample.log combined
```

RESUMEN DE OPERACIÓN DE FORMA DINÁMICA

CONFIGURACIÓN Y MÓDULOS ACTIVADOS

Utilizando los comandos `a2enmod` y `a2dismod` se pueden habilitar y deshabilitar los módulos, de la carpeta de módulos disponibles, se hace un link virtual en la carpeta de módulos habilitados. A continuación se presentan los módulos disponibles y los que fueron usados para esta prueba.

```
jair@lap-jair:/etc/apache2/mods-enabled$ sudo a2enmod
Your choices are: access_compat actions alias allowmethods asis auth_basic auth_digest auth_form authn_anon authn_core authn_dbd authn_dbm authn_
file authn_socache authnz_fcgi authnz_ldap authz_core authz_dbd authz_dbm authz_groupfile authz_host authz_owner authz_user autoindex buffer cach
e cache_disk cache_socache cern_meta cgi cgid charset_lite data dav dav_fs dav_lock dbd deflate dialup dir dump_io echo env expires ext_filter fi
le_cache filter headers heartbeat heartmonitor http2 ident imagemap include info lbmethod_bybusyness lbmethod_byrequests lbmethod_bytraffic lbmet
hod_heartbeat ldap log_debug log_forensic lua macro mime mime_magic mpm_event mpm_prefork mpm_worker negotiation php7.2 proxy proxy_ajp proxy_bal
ancer proxy_connect proxy_express proxy_fcgi proxy_fdpass proxy_ftp proxy_hcheck proxy_html proxy_http proxy_http2 proxy_scgi proxy_wstunnel rate
limit reflector remoteip reqtimeout request rewrite sed session session_cookie session_crypto session_dbd setenvif slotmem_plain slotmem_shm soca
che_dbm socache_memcache socache_shmcb spelling ssl status substitute suexec unique_id userdir usertrack vhost_alias xml2enc
Which module(s) do you want to enable (wildcards ok)?
```

```
jair@lap-jair:/etc/apache2/mods-enabled$ sudo a2dismod
Your choices are: access_compat alias auth_basic authn_core authn_file authz_core authz_host authz_user autoindex deflate dir env filter headers
mime mpm_prefork negotiation php7.2 reqtimeout setenvif socache_shmcb ssl status
Which module(s) do you want to disable (wildcards ok)?
```

RESUMEN DE OPERACIÓN DEL SERVIDOR

En status.conf se agregan las siguientes líneas para ver el server-status en apache.

```
# Uncomment and change the 192.0.2.1 to match the IP of your host  
  
<Location /server-status>  
    SetHandler server-status  
    <RequireAny>  
        Require local  
        Require ip 127.0.0.1  
    </RequireAny>  
</Location>
```

Dicha página, como se ve en las anteriores líneas, sólo puede ser visualizada por el propio servidor y da información útil acerca del uso de este. Datos como el porcentaje del procesador usado, las peticiones por segundo, etc.



Apache Server Status for www.example.com (via 127.0.1.1)

Server Version: Apache/2.4.29 (Ubuntu) OpenSSL/1.1.0g
Server MPM: prefork
Server Built: 2018-06-27T17:05:04

Current Time: Thursday, 20-Sep-2018 19:15:37 CDT
Restart Time: Thursday, 20-Sep-2018 16:29:31 CDT
Parent Server Config. Generation: 52
Parent Server MPM Generation: 51
Server uptime: 2 hours 46 minutes 5 seconds
Server load: 1.12 0.88 0.73
Total accesses: 305 - Total Traffic: 25.2 MB
CPU Usage: u.01 s0 cu0 cs0 - .0001% CPU load
.0306 requests/sec - 2653 B/second - 84.7 kB/request
1 requests currently being processed, 5 idle workers

.....
.....
.....

Scoreboard Key:

" " Waiting for Connection, "S" Starting up, "R" Reading Request,
"w" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	Protocol	VHost	Request
0-51	13291	0/0/131	_	0.00	17	6	0.0	0.00	1.38	::1	http/1.1		
1-51	13294	0/2/23	_	0.00	463	0	0.0	0.00	0.66	127.0.0.1	http/1.1	www.secdexample.com:80	GET /error_401.html HTTP/1.1
2-51	13293	0/2/19	W	0.00	0	0	0.0	0.00	10.56	127.0.0.1	http/1.1	www.example.com:80	GET /server-status HTTP/1.1
3-51	13292	0/2/35	_	0.00	17	0	0.0	0.00	0.59	::1	http/1.1		
4-51	13297	0/0/34	_	0.00	17	1	0.0	0.00	11.03	::1	http/1.1		
5-51	13290	0/1/40	_	0.00	417	2	0.0	0.00	0.78	127.0.0.1	http/1.1		
6-45	-	0/0/1	.	0.00	954	0	0.0	0.00	0.00	10.100.70.182	http/1.1		

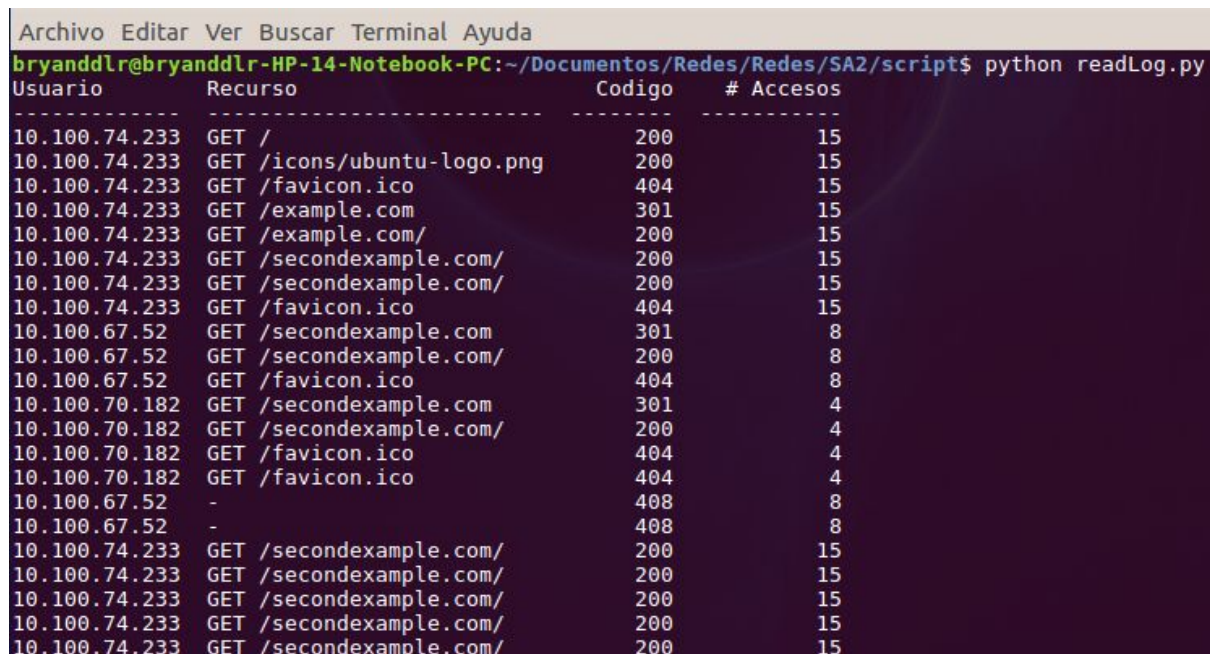
TABLAS DE RESÚMENES DE OPERACIÓN

Para los resúmenes de operación se tomó un formato de log personalizado, en el cuál se habilitaron las banderas necesarias para mostrar los datos requeridos.

Los scripts se generaron en pyhton 2.7, dado que el lenguaje brinda una facilidad en el manejo de ficheros (archivos .log) y facilidad al momento de parsear cada línea de los logs.

RESUMEN DE ERRORES DEL SERVIDOR

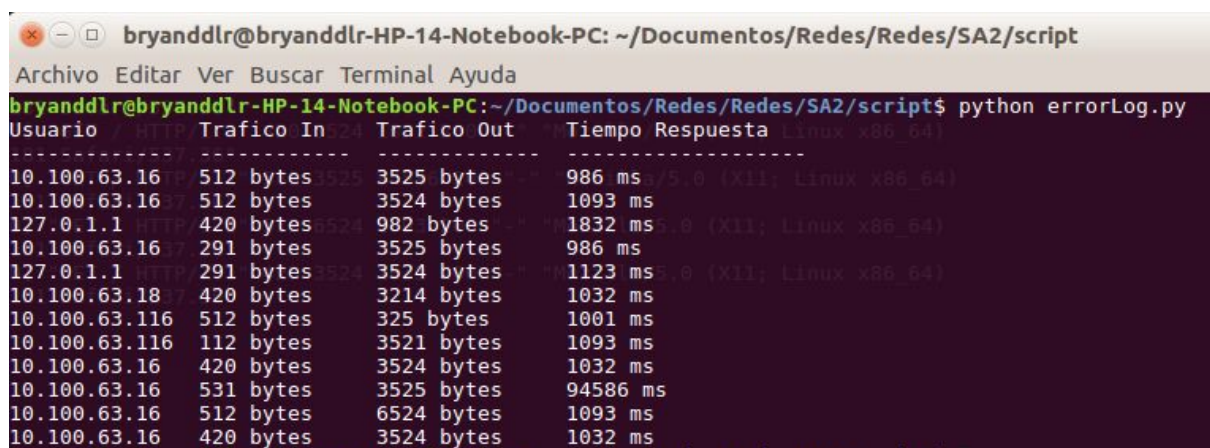
En este resumen se muestra una tabla con 4 columnas, las cuales muestran el usuario que realiza la petición, el recurso que está solicitando, el código de respuesta que brinda el servidor y el número de veces que un usuario ha realizado una petición.



Usuario	Recurso	Codigo	# Accesos
10.100.74.233	GET /	200	15
10.100.74.233	GET /icons/ubuntu-logo.png	200	15
10.100.74.233	GET /favicon.ico	404	15
10.100.74.233	GET /example.com	301	15
10.100.74.233	GET /example.com/	200	15
10.100.74.233	GET /seconddexample.com/	200	15
10.100.74.233	GET /seconddexample.com/	200	15
10.100.74.233	GET /favicon.ico	404	15
10.100.67.52	GET /seconddexample.com	301	8
10.100.67.52	GET /seconddexample.com/	200	8
10.100.67.52	GET /favicon.ico	404	8
10.100.70.182	GET /seconddexample.com	301	4
10.100.70.182	GET /seconddexample.com/	200	4
10.100.70.182	GET /favicon.ico	404	4
10.100.70.182	GET /favicon.ico	404	4
10.100.67.52	-	408	8
10.100.67.52	-	408	8
10.100.74.233	GET /seconddexample.com/	200	15
10.100.74.233	GET /seconddexample.com/	200	15
10.100.74.233	GET /seconddexample.com/	200	15
10.100.74.233	GET /seconddexample.com/	200	15
10.100.74.233	GET /seconddexample.com/	200	15

RESUMEN DE OPERACIÓN DEL SERVIDOR

En este resumen se muestra una tabla con 4 columnas, en la que se muestra el usuario que realizó la petición, los bytes que contenía su petición, los bytes de respuesta que le brindo el servidor y el tiempo de respuesta registrado por el servidor.



Usuario	HTTP Trafico In	Trafico Out	Tiempo Respuesta
10.100.63.16	512 bytes	3525 bytes	986 ms
10.100.63.16	512 bytes	3524 bytes	1093 ms
127.0.1.1	420 bytes	982 bytes	1832 ms
10.100.63.16	291 bytes	3525 bytes	986 ms
127.0.1.1	291 bytes	3524 bytes	1123 ms
10.100.63.18	420 bytes	3214 bytes	1032 ms
10.100.63.116	512 bytes	325 bytes	1001 ms
10.100.63.116	112 bytes	3521 bytes	1093 ms
10.100.63.16	420 bytes	3524 bytes	1032 ms
10.100.63.16	531 bytes	3525 bytes	94586 ms
10.100.63.16	512 bytes	6524 bytes	1093 ms
10.100.63.16	420 bytes	3524 bytes	1032 ms