

Programación en Perl orientada a la web



por
Matías Palomec
<matias@cafelug.org.ar>



Perl, CGI y HTTP



RFC 1945

Hypertext Transfer Protocol HTTP/1.0

0

El standar a respetar

RFC 1945

- Un standar reduce las probabilidades de hacer algo y que no funcione (aunque no lo elimina)
- Un standar abierto permite obtener todas las ventajas sin tener que pagar 60 USD (aprox.) por un PDF
- Respetar el standar facilita la portabilidad (entre otras muchas otras cosas)

Encabezados HTTP

- Un cliente pide una página web (usando telnet):

```
matias@nss:~$ telnet www.google.com 80
Trying 216.239.41.104...
Connected to www.google.akadns.net.
Escape character is '^]'.

```

```
GET http://www.google.com HTTP/1.0
```

Encabezados HTTP

- Y el servidor de google responde:

HTTP/1.0 302 Found

Date: Tue, 02 Nov 2004 07:11:01 GMT

Content-Length: 218

Content-Type: text/html

Location: [http://www.google.com/cx\[...\]](http://www.google.com/cx[...])

Encabezados HTTP

Para verificar los encabezados HTTP se puede utilizar la herramienta HEAD (parte del libwww de perl). En la forma:

```
HEAD http://dominio.tld/script.pl
```

Lo cual devolverá algo como lo anteriormente visto

Cuerpo del HTTP

El cuerpo de la petición HTTP
es la página HTML.

Que a su vez el HTML se divide en un
encabezado y en un cuerpo.

Perl

- Practical Extraction and Report Language (a.k.a. Perl)
- Es la panacea (y calvario) de los módulos (o sea, «alguien ya hizo un módulo que hace lo que necesito»)

Un script básico

```
#!/usr/bin/perl -w

print "Content-Type: text/html;charset=ISO-8859-15\n\n";

print <<EOP
<html>
<body>
<table width="100%" border="0"><tr valign="middle">
<td align="center">hola mundo</td></tr></table>
</body>
</html>
EOP

__END__
```

¿Qué hace eso?

- La primer línea es el SheBang
- La segunda, manda el header que le dirá al navegador que es una página HTML, y que utilice la tabla de caracteres ISO-8859-15 (español)
- La próxima línea imprime todo hasta que encuentra el “EOP”, o sea, un código HTML simple
- La última es el fin (todo lo que está despues no se interpreta)

Encabezados propios

También se pueden agregar encabezados propios, que no pertenecen a ningún RFC, y son para uso interno propio.

Poniendo algo como:

```
print "LUG: CaFeLUG\n";
```

¿qué mostraría eso?

```
200 OK
Connection: close
Date: Tue, 09 Nov 2004 13:10:22 GMT
Server: Apache/2.0.52
Content-Type: text/html;charset=ISO-8859-15
Client-Date: Tue, 09 Nov 2004 13:10:23 GMT
Client-Peer: 168.226.119.249:80
Client-Response-Num: 1
LUG: CaFeLUG
```

¿Y los módulos de Perl?

use CGI;

```
#!/usr/bin/perl -w

use CGI qw/-no_xhtml/;

my $q = new CGI;

print $q->header();
print $q->start_html(-title=>"Hola Mundo") . "\n";
print $q->table({-width=>"100%",
                -border=>"0"},
               $q->Tr({-valign=>"middle",
                    -align=>"center"},
                    [
                      $q->td({-valign=>"middle"},
                            ["Hola Mundo!!"]),
                    ]
                )
            );
print $q->end_html . "\n";
```

¿qué hace eso?

- SheBang
- Usa el módulo CGI y le dice al módulo que no use XHTML 1.0 (que es la opción por defecto)
- Crea un objeto llamado **\$q**
- Imprime por STDOUT el header (encabezado HTTP)
- Crea la página html y le da el título “Hola Mundo”
- Pone una tabla que es del ancho total y sin bordes, en dicha tabla pone una columna (alineada al medio), en donde pone una fila (alineada al centro) donde pone un “Hola Mundo!!”
- Termina la página html
- Los “\n” son opcionales, pero hacen más claro el código HTML resultante de este script

¿por qué usar CGI?

Una de las ventajas de utilizar el módulo CGI es que permite desligarse de escribir código HTML válido.

El módulo de CGI permite usar tanto la orientación a objetos, como las funciones.

Y sobre todo, facilita la utilización de los parámetros CGI

Parámetros CGI

Si se pasan valores al script, de forma:

<http://dominio.tld/script.pl?page=portada>

Yo para usar el valor de page puedo:

`$q->parma("page")`

o en la forma de función

`param("page")`

Lo anterior me devuelve "portada"

Sesiones en Perl

```
#!/usr/bin/perl -w

use CGI;
use CGI::Session;

my $q      = new CGI;
my $cookie = ""
my $session =
    new CGI::Session( 'driver:File', $q, { Directory => '/tmp/' } );

if ($q->cookie('CGISESSID')) {
    $cookie = $q->cookie('CGISESSID');
    $session->id($cookie);
} else {
    $cookie = $q->cookie(-name=>'CGISESSID',-value=>$session->id());
}

unless ($session->param('fecha')){
    $session->param(-name => 'fecha' , -value => scalar(gmtime()));
}

print $q->header(-cookie=>$cookie);
print $q->start_html("Ejemplo de manejo de sesiones");
print $q->h1("Cookie") . "\n" . $q->p($cookie);
print $q->h1("Fecha") . "\n" . $q->p("UTC: " . $session->param('fecha'));
print $q->end_html();
```

__END__

¿qué es lo que hace eso?

- Primero se cargan/usan los módulos CGI y CGI::Session, luego se inician las variables que se usarán (\$q, \$cookie y \$session).
- Se revisa si ya existe un cookie llamado 'CGISESSID' en el browser, y si existe se usa como sesión. Si no existe, se genera un nuevo identificador de sesión (en forma de cookie)
- Verifico que no exista una variable “fecha” dentro de la sesión, si no está la creo
- Muestro una página simple donde se ve el “Cookie” y la “Fecha”

¿y los templates?

El código del script

```
#!/usr/bin/perl -w

use HTML::Template;

my $template = HTML::Template->new(filename => 'ejemplo_template.tmpl');

$template->param("mensaje" => "Hola Mundo!!");

print "Content-Type: text/html;charset=UTF-8\n\n";
print $template->output;

__END__
```

El código del template:

```
<html>
<head>
<title>Template</title>
</head>

<body>

<h1>Mensaje desde el script</h1>

<p> <tmpl_var name=mensaje> </p>

</body>
</html>
```

Acceso abstracto a DB

Módulos:

DBI - Database independent interface for
Perl

Bases de datos disponibles para esde
módulo:

MySQL, PostgreSQL, CSV/TSV, ODBC,
Oracle, XBase, Interbase, Excel

Sitios recomendados

- <http://www.slashcode.com/> (el mismo de <http://barrapunto.com> y <http://slashdot.org>)
- <http://www.perl.org> y <http://www.perl.com>
- <http://cpan.perl.org> (Comprehensive Perl Archive Network)
- <http://groups.yahoo.com/group/perl-es/>

¿Dudas, preguntas?
¿Doubts, questions?
¿Duboj, demandoj?