REDES DE COMPUTADORES Y LABORATORIO

Christian Camilo Urcuqui López, MSc





BIBLIOGRAFÍA











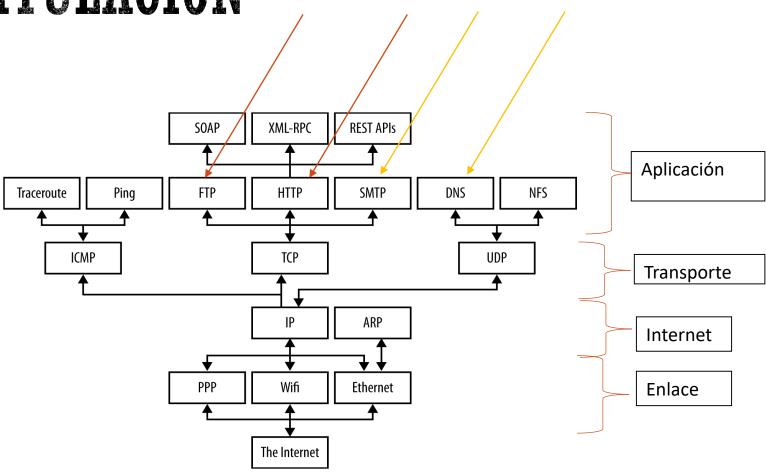


COMPETENCIAS

- Describir la capa de aplicación.
 - Describir World Wide Web
 - Páginas estáticas
 - Páginas dinámicas



RECAPITULACIÓN





LA CAPA DE APLICACIÓN

CSS: CASCADING STYLE SHEETS

• El objetivo original de HTML era especificar la **estructura** del documento, no su **apariencia**.

Fotos de
Débora

• En los editores de texto, las hojas de estilo permiten a los autores asociar texto con un estilo lógico en vez de uno físico; por ejemplo, "párrafo inicial" en vez de "texto en cursiva". La apariencia de cada estilo se define por separado. De esta manera, si el autor decide cambiar los párrafos iniciales de cursiva a 14 puntos en color azul, a negritas a 18 puntos en rosa impactante, todo lo que requiere es modificar una definición para convertir todo el documento.

CSS: CASCADING STYLE SHEETS

 CSS define un lenguaje simple para describir reglas que controlan la apariencia de contenido etiquetado

```
body {background-color:linen; color:navy; font-family:Arial;} h1 {font-size:200%} h2 {font-size:150%}
```

Figura 7-27. Ejemplo de CSS.

 Podemos colocar las hojas de estilo en un archivo HTML (mediante la etiqueta <style>), pero es más común colocarlas en un archivo separado y hacer referencia a ellas.

```
<title>WIDGETS AMALGAMADOS, S.A. </title>
k rel="stylesheet" type="text/css" href= "awistyle.css" />
</head>

MIME
```

• El contenido dinámico se puede generar mediante programas que se ejecuten en el servidor o en el navegador (o en ambos lugares).

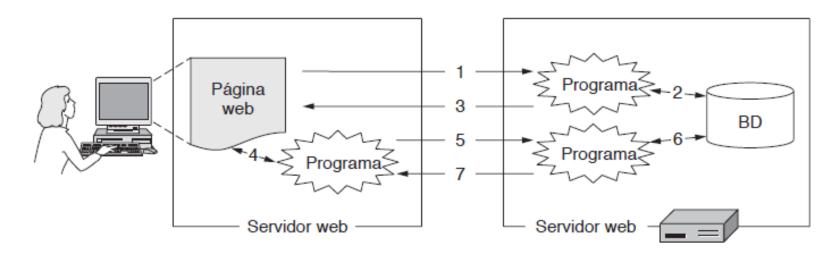


Figura 7-29. Páginas dinámicas.

Se han desarrollado varias API estándar para que los servidores web invoquen programas. La existencia de estas interfaces facilita a los desarrolladores el proceso de extender distintos servidores con aplicaciones web. A continuación analizaremos dos API para que el lector se dé una idea de lo que implican.

1. CGI (Interfaz de Puerta de Enlace Común, del inglés Common Gateway Interface) y se define en el RFC 3875. CGI provee una interfaz para permitir que los servidores web se comuniquen con programas de soporte y secuencias de comandos que pueden aceptar entradas (por ejemplo, de los formularios) y generar páginas HTML en respuesta.

Estos programas pueden estar escritos en cualquier lenguaje que sea conveniente para el desarrollador, por lo general un lenguaje de secuencias de comandos para facilitar el avance. Puede elegir Python, Ruby, Perl o su lenguaje favorito.

Por convención, los programas que se invocan a través de CGI viven en un directorio llamado cgibin, el cual es visible en el URL.

- 2. La metodología en este caso es incrustar pequeñas secuencias de comandos dentro de las páginas de HTML y hacer que las ejecute el mismo servidor para generar la página. Un lenguaje popular para escribir estas secuencias de comandos es **PHP** (**Preprocessor**).

 (Preprocessor).
 - El servidor tiene que entender PHP. Por lo general, los servidores identifican las páginas web que contienen PHP con base en la extensión de archivo *php*.

```
<html>
<body>
<form action="accion.php" method="post">
 Por favor introduzca su nombre: <input type="text" name="nombre"> 
 Por favor introduzca su edad: <input type="text" name="edad"> 
<input type="submit">
</form>
</body>
</html>
                                                  (a)
<html>
<body>
<h1> Respuesta: </h1>
Hola <?php echo $nombre; ?>.
Predicción: el siguiente año tendrá <?php echo $edad + 1; ?>
</body>
</html>
                                                  (b)
<html>
<body>
<h1> Respuesta: </h1>
Hola Bárbara.
Predicción: el siguiente año tendrá 33
</body>
</html>
                                                  (c)
```

Figura 7-30. (a) Una página web que contiene un formulario. (b) Una secuencia de comandos de PHP para manejar la salida del formulario. (c) Salida de la secuencia de comandos de PHP cuando las entradas son "Bárbara" y "32", respectivamente.

- Existen otras opciones disponibles:
- JSP (JavaServer Pages) la parte dinámica se escribe en el lenguaje de programación Java. Las páginas tienen la extensión de archivo .jsp.
- **ASP.NET** (*Active Server Pages .NET*). Es la versión de Microsoft que utiliza programas escritos en el framework en red .NET para generar el contenido dinámico. Las páginas escritas en esta tecnología tienen extensión .aspx.



Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C++		99.7
3. Java		97.5
4. C		96.7
5. C#		89.4
6. PHP		84.9
7. R		82.9
8. JavaScript		82.6
9. Go	● □	76.4
10. Assembly		74.1

IEEE ESPECTRUM - The 2018 Top Programming Languages. https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages

GENERACIÓN DE PÁGINAS WEB DINÁMICAS EN EL CLIENTE

- Lo que ninguno de los lenguajes previamente mencionados puede realizar es responder a interacciones directas con el usuario (por ejemplo, los movimientos del ratón).
- Para ello es necesario tener secuencias de comandos incrustados en páginas HTML que se ejecuten en la máquina del cliente y no en el servidor.
- Los comandos se pueden ingresar mediante la etiqueta <script> y las páginas web interactivas se conocen como **HTML dinámico**.
- JavaScript es un lenguaje de alto nivel para el desarrollo de páginas interactivas.



```
<html>
<head>
<script language="javascript" type="text/javascript">
function respuesta(form_prueba) {
    var persona = form_prueba.nombre.value;
    var anios = eval(form_prueba.edad.value) + 1;
    document.open();
    document.writeln("<html> <body>");
    document.writeln("Hola " + persona + "<br>");
    document.writeln("Predicción: el siguiente año tendrá " + anios + ");
    document.writeln("</body> </html>");
    document.close();
</script>
</head>
<body>
<form>
Por favor introduzca su nombre: <input type="text" name="nombre">
>
Por favor introduzca su edad: <input type="text" name="edad"> 
>
<input type="button" value="submit" onclick="respuesta(this.form)">
</form>
</body>
</html>
```

Figura 7-31. Uso de JavaScript para procesar un formulario.

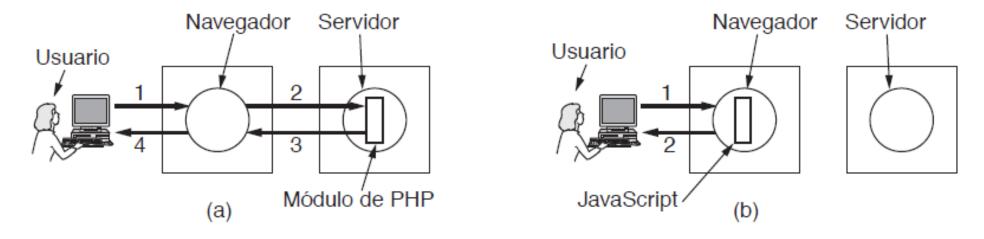
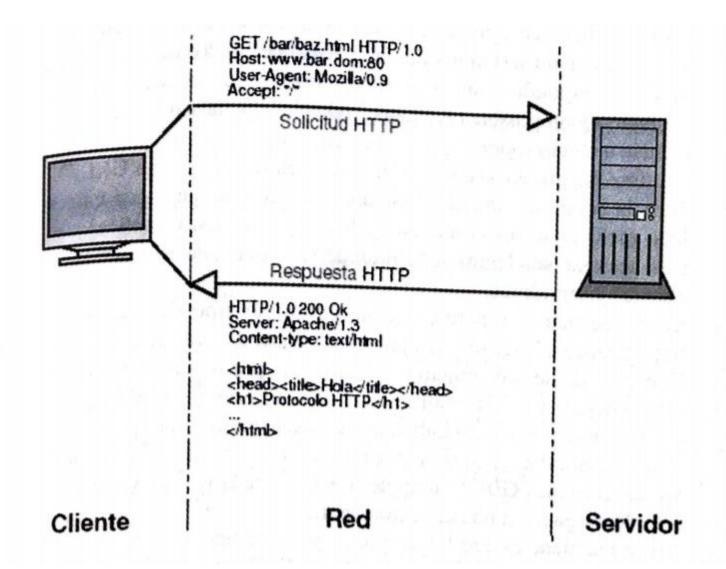


Figura 7-32. (a) Secuencias de comandos del lado servidor con PHP. (b) Secuencias de comandos del lado cliente con JavaScript.

Otras alternativas a JavaScript:

- En las plataformas Windows es VBScript, que se basa en Visual Basic.
- Applets, son pequeños programas en Java compilados en instrucciones de máquina para una computadora virtual, conocida como JVM (Java Virtual Machine)



HTTP (HYPERTEXT TRANSFER PROTOCOL)

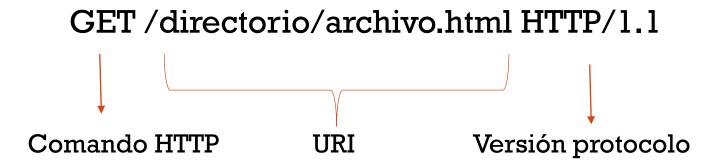
Es un protocolo de nivel de aplicación que corre sobre TCP/IP y se usa por clientes y servidores Web para la transferencia de datos multimedia en forma de hipertexto a través de hipervínculos.

El servidor Web acepta conexiones HTTP (usualmente enviadas desde los browsers de los usuarios), busca la información solicitada en un URL en el sistema de archivos local y la devuelve al cliente.



ENVIO DE SOLICITUD HTTP

- Las solicitudes y las respuestas tienen las mismas partes:
 - Línea inicial
 - Cabeceras de solicitud
 - Cero o más campos de encabezado
 - Dos líneas vacías (CR/CL)
 - Puede incluir el cuerpo del mensaje



ENVIO DE SOLICITUD HTTP

Cabeceras de solicitud:

 Proveen información adicional sobre la solicitud y la respuesta. El formato es "nombre": "valor"

▼ General

Request URL: https://www.icesi.edu.co/

Request Method: GET

Status Code: • 200 OK

Remote Address: 200.3.192.46:443

Referrer Policy: no-referrer-when-downgrade

HTTP (HYPERTEXT TRANSFER PROTOCOL)

- No es un protocolo orientado a conexión, es decir, los clientes y el servidor no tienen una conexión permanente (como sí ocurre en protocolos como FTP), sino que se comunican a intervalos por medio de mensajes solicitud-respuesta inmediatamente después pierde la conexión.
- El protocolo HTTP no guarda estado, es trabajo del servidor guardar información sobre el proceso de cada cliente.
- Cada paquete HTTP funciona de manera similar al resto de servicios TCP/IP, un proceso servidor que escucha sobre el puerto 80.



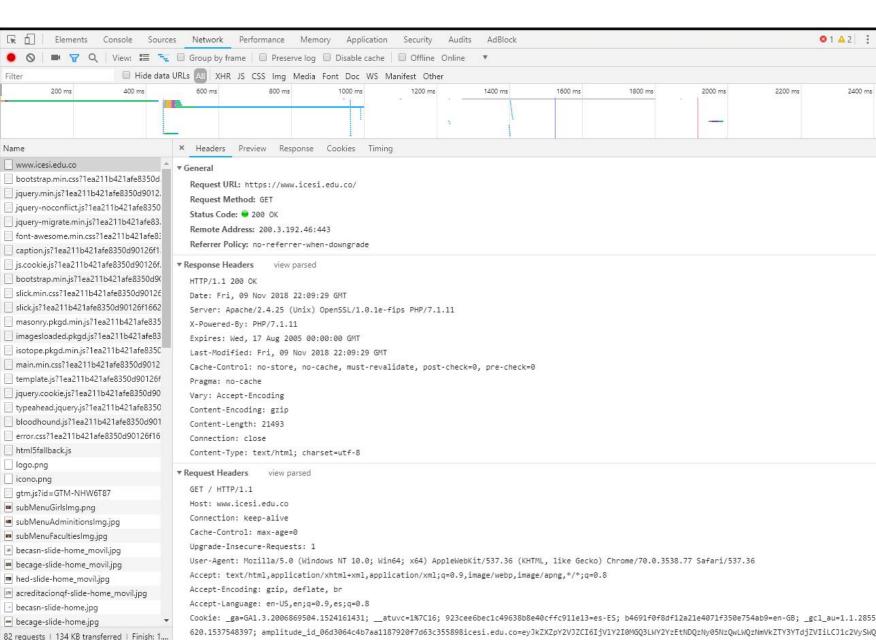


¿Qué programa estás buscando?

Noticias y Eventos



EVENTO Club De Lectura



◎1 ▲2 :

2400 ms

2200 ms

```
Request URL: https://www.icesi.edu.co/
   Request Method: GET
   Status Code: 9 200 OK
   Remote Address: 200.3.192.46:443
   Referrer Policy: no-referrer-when-downgrade
▼ Response Headers
                     view parsed
   HTTP/1.1 200 OK
   Date: Fri, 09 Nov 2018 22:09:29 GMT
   Server: Apache/2.4.25 (Unix) OpenSSL/1.0.1e-fips PHP/7.1.11
   X-Powered-By: PHP/7.1.11
   Expires: Wed, 17 Aug 2005 00:00:00 GMT
  Last-Modified: Fri, 09 Nov 2018 22:09:29 GMT
  Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
   Pragma: no-cache
  Vary: Accept-Encoding
  Content-Encoding: gzip
   Content-Length: 21493
   Connection: close
   Content-Type: text/html; charset=utf-8
▼ Request Headers
                   view parsed
   GET / HTTP/1.1
   Host: www.icesi.edu.co
   Connection: keep-alive
   Cache-Control: max-age=0
   Upgrade-Insecure-Requests: 1
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.77 Safari/537.36
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/appg,*/*;q=0.8
  Accept-Encoding: gzip, deflate, br
   Accept-Language: en-US,en;q=0.9,es;q=0.8
  Cookie: _ga=GA1.3.2006869504.1524161431; __atuvc=1%7C16; 923cee6bec1c49638b8e40cffc911e13=es-ES; b4691f0f8df12a21e4071f350e754ab9=en-
   620.1537548397; amplitude_id_06d3064c4b7aa1187920f7d63c355898icesi.edu.co=eyJkZXZpY2VJZCI6IjV1Y2I0MGQ3LWY2YzEtNDQzNy05NzQwLWQzNmVkZTY
```

EALS TO JEGA STORE OF THE ON THE OPERATOR TRANSPORTED TO USE JEGA DESCRIPTION OF TRANSPORTED TO USE A PROPERTY OF THE PROPERTY

▼ General

MÉTODOS

Método	Descripción	
GET	Leer una página web.	
HEAD	Leer el encabezado de una página web.	
POST	Adjuntar a una página web.	
PUT	Almacenar una página web.	
DELETE	Eliminar la página web.	
TRACE	Repetir la solicitud entrante	
CONNECT	Conectarse a través de un proxy	
OPTIONS	Consultar las opciones para una página	

Figura 7-37. Los métodos de solicitud HTTP integrados.

▼ General

Request URL: https://www.icesi.edu.co/

Request Method: GET

Status Code: 9 200 OK

Remote Address: 200.3.192.46:443

Referrer Policy: no-referrer-when-downgrade

Código	Significado	Ejemplos
1xx	Información	100 = el servidor acepta manejar la solicitud del cliente.
2xx	Éxito	200 = la solicitud es exitosa; 204 = no hay contenido.
3xx	Redirección	301 = se movió la página; 304 = la página en caché aún es válida.
4xx	Error del cliente	403 = página prohibida; 404 = no se encontró la página.
5xx	Error del servidor	500 = error interno del servidor; 503 = intentar más tarde.

Figura 7-38. Los grupos de respuesta del código de estado.

▼ General

Request URL: https://www.icesi.edu.co/

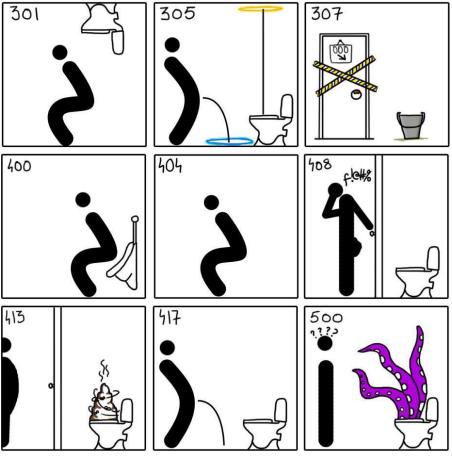
Request Method: GET

Status Code: © 200 OK

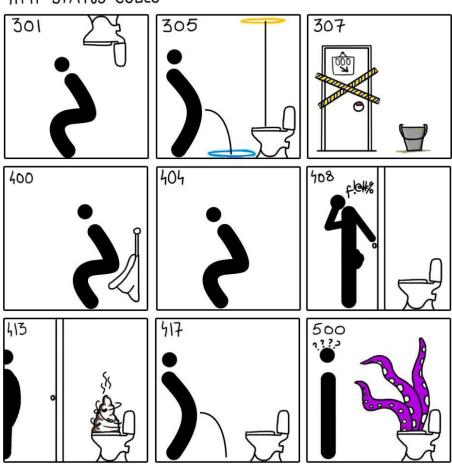
Remote Address: 200.3.192.46:443

Referrer Policy: no-referrer-when-downgrade

HTTP STATUS CODES



HTTP STATUS CODES



301: Se movió la página

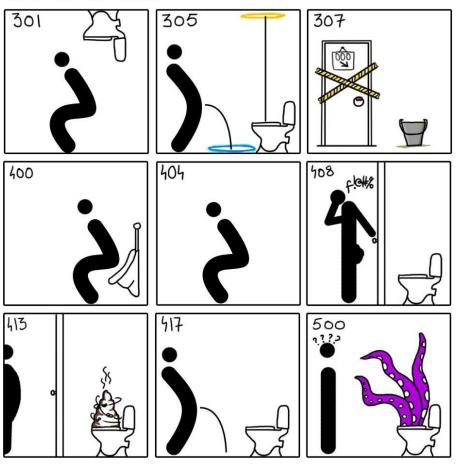
305: Uso de Proxy

307: Temporalmente re direccionado

400: Mala solicitud

404: No se encontró la página

HTTP STATUS CODES



408: El cliente fallo en la petición

413: La petición es demasiado grande

417: La petición falla, el servidor no es capaz de procesar los requerimientos

500: Error interno del servidor





500. That's an error.

There was an error. Please try again later. That's all we know.



500 Internal Server Error

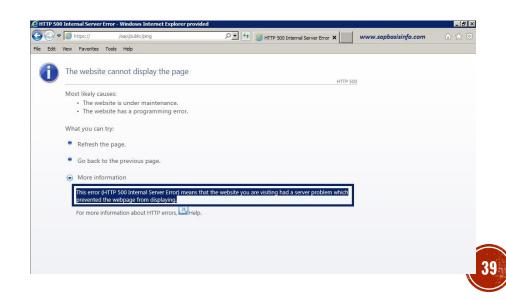
Sorry, something went wrong.

A team of highly trained monkeys has been dispatched to deal with this situation.

If you see them, send them this information as text (screenshots frighten them):

APkpgMWo53Cv7b61RrV5UpKVjyi5d4S8UcSGX-jKPXheRLc0XfbJIy5e hEbf09icypEN9x1Au4TKa-7Bf0fFD4 tH9HltJ200wJ2UIKcv2VSsEhD izrxrP BuSP-KScpQlrpqQGBJbopRnLS634a0wSB6QBPViv87QSZ4Plz M1K6-KOpV37SMxThqa9 i7oYS5nq8lGi81VyLY3StYtr9RxyrjlE6 41 O494rGRINzHXk7aJCMOctmN23 E6X0PSd8-JN4Ya9xNGqvcLNaZKzNU7 6ufis8 nPF8o5Yr-Gsjh-yFEy1-u4YUbtf94Vw0U02b5GGbewGz16jXc 2IMs1602cho2Us18S6bKfDKBXo8TCeWud7dMUjWJQDc10LQuxhnUZ2M1 STuUmaxlCjsQelCiQjhdG-u0mxztdbQss6M-ocXI3z7kSJdysRE5azBP N69bhFR931K4DkVjMXZxwlANcz6a5Z2RjRRkQnAZl_iGYqTGdn40gSYn WQiiRtIFyHPuW87yzp_kTPI56L9adx_95JUb7aNYn5CBHGcfoeistG15 Daa_3lq4menUbOfIz7k7lSiMB0Rli7ZrHcYVAvnMWVsgf-IxKbeb1t23 sckK7Ss6o52L0o0109qCtoNq b7apXx 6q8dNozfVb0oaPnu5TlAvUW9 YqEdpuNVvYr-8dcwnm013WcC4g9w22q0wkk7WUnizAYdHmRXgCKMxwzH xF9hmBly4q6LA7bRpj-CTaJxUnuE03 ZCkBjvV-fqDcMQJFNwQ o7JVO RDdn3UqSjaaqoDjpACtBlUr9OIoAOrHavlzKFvmLHWXzhXbq-zj oCSc He6z8ixm5ma32-0oAqAhvcCTkPAEayeT2yDXo6qaG1TwC0he5weUjKE1 GwtoJvlTYJYxpBX2zemYV5s8xvO0cyH2VimbhG4tKpGmubLUDkT wEWi OgL3m51-WPhpGXfpkRZKt8BjTLVUrklIjSrSOsgiUZb2xnVIknb51ABm





GET / HTTP/1.1

Host: www.icesi.edu.co
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.77 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.9,es;q=0.8

Cookie: _ga=GA1.3.2006869504.1524161431; __atuvc=1%7C16; 923cee6bec1c49638b8e40cffc911e13=es-ES; b4691f0f8df12a21e4071f350e754ab9=en-620.1537548397; amplitude_id_06d3064c4b7aa1187920f7d63c355898icesi.edu.co=eyJkZXZpY2VJZCI6IjV1Y2I0MGQ3LWY2YzEtNDQzNy05NzQwLWQzNmVkZT\

FALS TO USON LETGE S. OU T. NO. ON LOSS SECURITY OFFICENTION. "LES USUADISCUSAL SECURITY OFFICENTION. "SAYS". BOSECUSE " USUADIS SECURITY OF SECURITY

Encabezado	Tipo	Contenidos
User-Agent	Solicitud	Información sobre el navegador y su plataforma.
Accept	Solicitud	El tipo de páginas que puede manejar el cliente.
Accept-Charset	Solicitud	Los conjuntos de caracteres que son aceptables para el cliente.
Accept-Encoding	Solicitud	Las codificaciones de página que puede manejar el cliente.
Accept-Language	Solicitud	Los idiomas naturales que puede manejar el cliente.
If-Modified-Since	Solicitud	Hora y fecha para verificar la actualidad de un mensaje.
If-None-Match	Solicitud	Etiquetas enviadas previamente para verificar la actualidad de un mensaje.
Host	Solicitud	El nombre DNS del servidor.
Authorization	Solicitud	Una lista de las credenciales del cliente.
Referer	Solicitud	El URL anterior desde el cual provino la solicitud.
Cookie	Solicitud	La cookie establecida previamente que se regresa al servidor.

▼ Response Headers view parsed

HTTP/1.1 200 OK

Date: Fri, 09 Nov 2018 22:09:29 GMT

Server: Apache/2.4.25 (Unix) OpenSSL/1.0.1e-fips PHP/7.1.11

X-Powered-By: PHP/7.1.11

Expires: Wed, 17 Aug 2005 00:00:00 GMT

Last-Modified: Fri, 09 Nov 2018 22:09:29 GMT

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

Pragma: no-cache

Vary: Accept-Encoding Content-Encoding: gzip Content-Length: 21493

Connection: close

Content-Type: text/html; charset=utf-8

Content-Range	Respuesta	Identifica una parte del contenido de la página.
Last-Modified	Respuesta	Hora y fecha de la última modificación de la página.
Expires	Respuesta	Hora y fecha en que la página dejará de ser válida.
Location	Respuesta	Indica al cliente a dónde enviar su solicitud.
Accept-Ranges	Respuesta	Indica que el servidor aceptará solicitudes de rango de bytes.
Date	Ambas	Fecha y hora en que se envió el mensaje.
Range	Ambas	Identifica una parte de una página.
Cache-Control	Ambas	Directivas para manejar las cachés.
ETag	Ambas	Etiqueta para el contenido de la página.
Upgrade	Ambas	El protocolo al que el emisor desea conmutar.

PLANEACIÓN

Material utilizado	1. Arboleda, L. (2012). Programación en Red con Java. 2. Harold, E. (2004). Java network programming. " O'Reilly Media, Inc.". 3. Tanenbaum, A. S. (2003). Redes de computadoras. Pearson educación. 4. Reese, R. M. (2015). Learning Network Programming with Java. Packt Publishing Ltd.
Actividades DESPUÉS clase	A3. Leer la sección 15 del libro 1 y Leer del libro 4 la sección 4



REFERENCIAS

- https://upload.wikimedia.org/wikipedia/commons/thumb/4/4e/Sir Tim Berners-Lee %28cropped%29.jpg
- https://upload.wikimedia.org/wikipedia/commons/thumb/7/75/Marc_Andreessen_jpg/200px-Marc_Andreessen.jpg
- https://commons.wikimedia.org/wiki/File:URI Venn Diagram.svg
- https://qph.fs.quoracdn.net/main-qimglce27ba8ele52a0aa7889722295lc538.webp
- https://www.optimisthub.com/up/frontend-nedir.png
- https://dev.to/flippedcoding/what-is-the-difference-between-a-uri-and-a-url-4455