

Laboratorio de Desarrollo de Aplicaciones Web

TC3052

Jorge Miramontes Amezcua

Project Lead, P3 Group

Profesor de Cátedra ITESM – CEM

jmiramontes@tec.mx

Agenda

Reglas

Temario

Dinámica de la clase

Presentación Individual

Protocolo HTTP

Práctica 1 Grupal

Práctica 1 Individual

Reglas

Pase de Lista 19:05 horas.

10 minutos de tolerancia. Después de las 19:15 horas no se permitirá la entrada al salón por ningún motivo.

El uso de dispositivos electrónicos (celular, Tablet, laptop, etc.) está limitado a las necesidades del laboratorio.

Reglas

Contacto y Asesoría

Google Classroom: Invitación por correo electrónico

Correo Electrónico: jmiramontes@tec.mx

Medio de comunicación oficial.

Reglas

Evaluación Parcial: 2 Parciales

Prácticas Semanales	100%
---------------------	------

Evaluación Final

Primer Parcial	20%
Segundo Parcial	20%
Semana i	5%
Prácticas Individuales Finales	10%
Práctica Final	45%
Total	100%

Reglas

Prácticas Individuales / Equipo:

Si la práctica no está en Google Classroom, en la actividad correspondiente, no se revisará ni calificará.

Se revisará y se calificará sobre 100, como máximo una sesión posterior a la que se dejó la práctica.

Se revisará y calificará sobre 80, la siguiente sesión, aunque la práctica se haya subido a Google Classroom antes de esta sesión.

Posterior a esas sesiones, la calificación será cero, aunque la práctica se haya subido a Google Classroom antes de esta sesión.

Hay prácticas nuevas en cada sesión. **Ojo con las faltas.**

En caso de faltar a una sesión, es responsabilidad del alumno ponerse al corriente para poder entregar las prácticas en tiempo y forma. En la asesoría sólo se aclaran dudas muy específicas.

Los criterios de evaluación de las prácticas se especificarán en cada una.

Ejemplo:

Sesión	Se califica sobre
1ra sesión (en la que se dejó la práctica)	100
2da sesión	100
3ra sesión	80
4ta sesión y posteriores	La calificación será cero

Reglas

Práctica Final:

En equipo.

La presentación final será la última sesión. De ser necesario se utilizará el día del examen final para revisión.

La calificación estará en función de las soluciones presentadas y las preguntas respondidas en la presentación.

Formar Equipos

10 equipos de 2 personas

1 equipo de 3 personas

Enviar por correo a jmiramontes@tec.mx

Nombre del Equipo

Nombres y matrículas de los integrantes

Temario

Protocolo HTTP

Servicios Web REST

MEAN Stack

- MongoDB
- Express
- Angular / Bootstrap
- NodeJS
- Ionic

Lenguajes de Programación / Bases de Datos	Frameworks / APIs	Herramientas de Desarrollo / Servidores Web
JavaScript	NodeJS	Atom
TypeScript	Express	Postman
MongoDB	REST Servicios Web y Clientes	MongoDB Compass
	Angular	Wireshark
	Bootstrap	
	Ionic	

Dinámica de la clase

LABORATORIO

Teoría

Prácticas Grupales e Individuales

Presentación Individual

Nombre completo

Experiencia desarrollando Aplicaciones Web (aplicaciones, lenguajes, herramientas)

Experiencias / Expectativas Laborales

Protocolo HTTP

¿Qué es HTTP?

¿Qué es un “recurso”?

Estructura de una transacción HTTP

Métodos

Ejemplo de una transacción

Cientes HTTP

Servidores HTTP

Protocolo HTTP

¿Qué es HTTP?

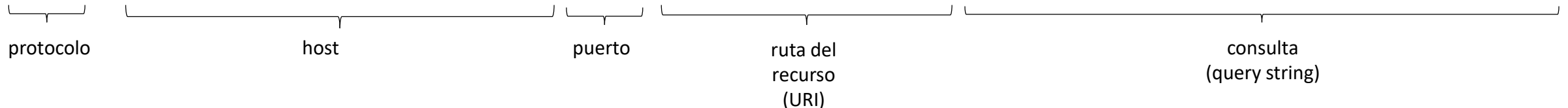
- HyperText Transfer Protocol.
- Es un protocolo de comunicación entre sistemas distribuidos. Acceso a “recursos” en sistemas remotos.
- La comunicación se realiza principalmente por medio de TCP / IP, pero se puede utilizar otro protocolo de transporte.
- Puerto por default: 80. Se puede utilizar otro puerto.
- RFC 7230 al RFC 7240 (<https://tools.ietf.org/html/rfc7230>)

Protocolo HTTP

¿Qué es un “recurso”?

- Información que puede ser identificada por medio de una URL (Uniform **Resource** Locator).
- Un recurso puede ser:
 - Archivo texto.
 - Imagen.
 - Resultado de una consulta a una base de datos.
 - Vídeo.
 - Sonido.
 - Etc.

- `http://www.midominio.com.mx:5318/ruta/del/recurso?variable1=valor1&variable2=valor2`



Protocolo HTTP

Estructura de una transacción HTTP

El formato de mensajes de un request y un response son similares, y constan de:

ELEMENTOS

- Línea inicial
- Cero o más líneas de encabezado (header)
- Línea en blanco (CRLF -> \r\n)
- Cuerpo del mensaje (opcional) (archivo, imagen, etc.)

ESTRUCTURA

Línea inicial <diferente para req y resp>

Header1: value1

Header2: value2

Header3: value3

<cuerpo opcional del mensaje>

Las líneas iniciales y los encabezados deben terminar con CRLF.

Protocolo HTTP

Estructura de una transacción HTTP

Línea inicial

- Es diferente para request y response.
- Request: Consta de 3 partes:
 - Separadas por espacio.
 - Nombre del método.
 - Ruta local del recurso.
 - Versión de HTTP

GET /ruta/del/recurso/img.jpg HTTP/1.1

Protocolo HTTP

Estructura de una transacción HTTP

- Response: También se le conoce como “status line”. Consta de 3 partes:
 - Versión HTTP.
 - Código de estatus de la respuesta.
 - Descripción del código de estatus de la respuesta.

HTTP/1.1 200 OK

HTTP/1.1 404 Not Found

- El código de estatus es un entero de 3 dígitos. El primer dígito sirve para identificar la categoría de la respuesta:

1xx: Mensaje informativo (Ejemplo: Continue (100), Switching Protocols (101)).

2xx: Mensaje de éxito (Ejemplo: OK (200), Accepted (202)).

3xx: Redirección a otra URL (Ejemplo: Moved Permanently (301)).

4xx: Error por parte del cliente (Ejemplo: Bad Request (400), Forbidden (401), Not Found (404)).

5xx: Error por parte del servidor (Ejemplo: Internal Server Error (500), Not Implemented (501)).

- Códigos de Respuesta: <http://tools.ietf.org/html/rfc7231#page-47>

Protocolo HTTP

Estructura de una transacción HTTP

Líneas de Encabezado

- Información acerca del request, response, o del contenido del cuerpo del mensaje.
- Una línea por encabezado, en la forma “Header-Name: value” y finalizando con CRLF. Se puede tener cualquier cantidad de espacios o tabuladores entre ‘:’ y ‘value’.
- Se pueden tener múltiples ‘value’ separados por coma. Si se utilizan múltiples ‘value’, también se pueden expresar en una nueva línea cada uno para facilitar su lectura.
- Header1: valor1, valor2
- Header1: valor1,
valor2
- HTTP 1.1 define 46 encabezados (HTTP 1.0 definía 16). El encabezado Host: es el único que es requerido para los mensajes de tipo request.

Protocolo HTTP

Estructura de una transacción HTTP

- En un request, se puede utilizar el encabezado 'User-Agent'. Define el programa que está haciendo el request "Nombre-Programa/x.xx". Por ejemplo: User-Agent: Mozilla/5.0
- En un response, se puede utilizar el encabezado 'Server'. Es Análogo a 'User-Agent'. Por ejemplo: Server: Aapache/3.1.1.
- En un response, se recomienda utilizar el encabezado 'Last-Modified'. Provee la fecha de modificación del recurso que se está regresando. Usado para caching y algoritmos que ahorran en el consumo de ancho de banda. Por ejemplo: "Last-Modified: Fri, 31 Dec 2014 17:35:21 GMT".
- ¿Podría ser contraproducente utilizar los encabezados 'User-Agent' y 'Server'?
- Encabezados de Request: <http://tools.ietf.org/html/rfc7231#page-33>
- Encabezados de Response: <http://tools.ietf.org/html/rfc7231#page-64>

Protocolo HTTP

Estructura de una transacción HTTP

Cuerpo del Mensaje

- Es opcional.
- Request: Contiene información del formulario que llenó el usuario, el contenido de un archivo que requiere subir al servidor.
- Response: Contenido del recurso solicitado por el cliente.
- Se recomienda utilizar los siguientes encabezados:
 - Content-Type: Tipo de contenido que contiene el encabezado (<http://tools.ietf.org/html/rfc2046#page-6>)
 - text/html (plain, javascript)
 - application/octet-stream (pdf, powerpoint)
 - image/jpg (gif, png)
 - Content-Length: Tamaño, en bytes, del cuerpo del mensaje. Útil para descarga de archivos, conocer el tamaño del archivo que se está descargando.

Protocolo HTTP

Estructura de una transacción HTTP

Métodos

- Son utilizados únicamente en el request.
- Acción que se quiere realizar sobre el recurso solicitado.
- Los más comunes son:
- GET: Solicita información sobre un recurso.
 - Se pueden incluir parámetros de consulta en el URI:
<http://www.midominio.com.mx:5318/ruta/del/recurso?variable1=valor1&variable2=valor2>
 - Limitado en capacidad de información a enviar. Depende del servidor el tamaño máximo. No es práctico el upload de archivos.
 - La URL se puede colocar como bookmark, facilitando al usuario consultas posteriores.
 - Inseguro ya que la información está contenida en el URI como texto plano.
- POST: Solicita información sobre un recurso. Principalmente mediante la invocación de un script en el servidor.
 - <http://www.midominio.com.mx:5318/ruta/del/script/altaUsuario.faces>
 - No está limitado en capacidad de información a enviar. Práctico para upload de archivos.
 - No se puede colocar la URL como bookmark. No tiene sentido.
 - Seguro ya que la información está contenida dentro del cuerpo del mensaje y no es visible en la URI. (Por ejemplo: usuario y contraseña).

Protocolo HTTP

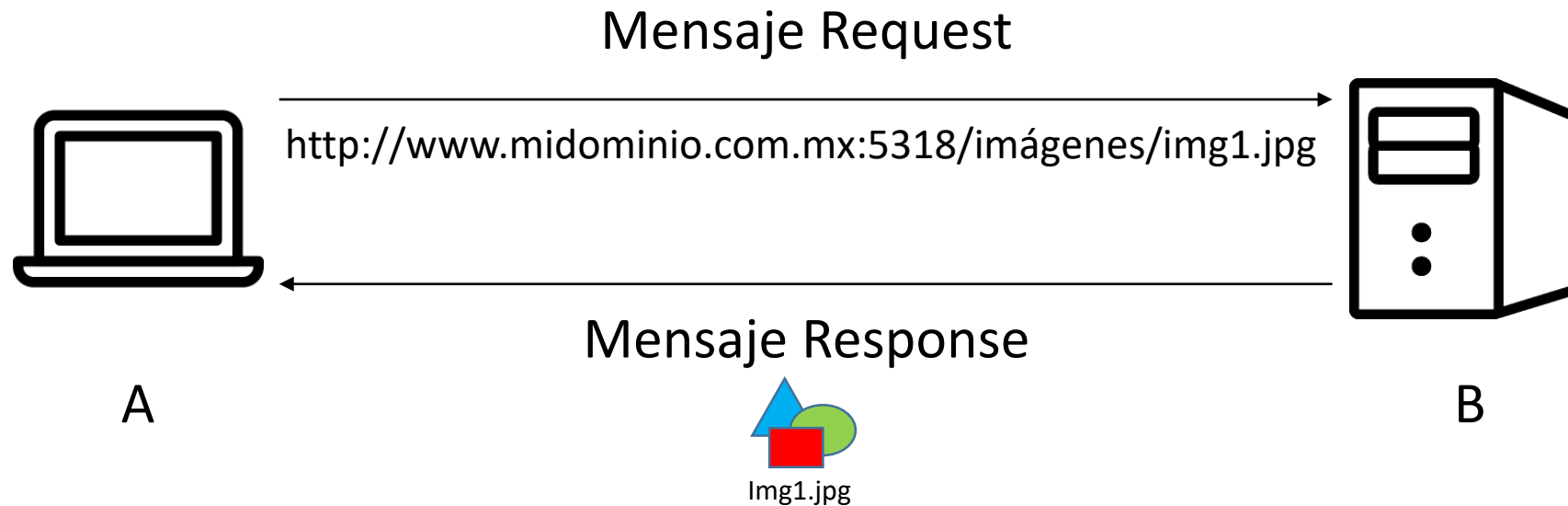
Estructura de una transacción HTTP

- PUT: Crea o actualiza un recurso.
- DELETE: Elimina un recurso existente.
- HEAD: Obtener el mensaje sin el cuerpo del mensaje, solamente los encabezados. Utilizado para revisar el estado de la información del recurso, por ejemplo, si ha sido modificado.
- OPTIONS: Consultar los métodos soportados por la URL especificada.

Protocolo HTTP

Ejemplo de una transacción

Ejemplo



Protocolo HTTP

Ejemplo de una transacción

Ejemplo

Mensaje Request

▼ Hypertext Transfer Protocol

> GET /sc/x/default/2015/09/01/00121441132363713799857/Foto/2013.jpg HTTP/1.1\r\n

Línea inicial

Host: www.lavozdegalicia.es\r\n

Connection: keep-alive\r\n

Cache-Control: max-age=0\r\n

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

Upgrade-Insecure-Requests: 1\r\n

Encabezados

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36\r\n

Accept-Encoding: gzip, deflate, sdch\r\n

Accept-Language: en-US,en;q=0.8,es;q=0.6,gl;q=0.4\r\n

If-None-Match: "e3d761-7ff5-51eb3c771eff7"\r\n

If-Modified-Since: Tue, 01 Sep 2015 18:32:57 GMT\r\n

\r\n

Línea en blanco

[Full request URI: <http://www.lavozdegalicia.es/sc/x/default/2015/09/01/00121441132363713799857/Foto/2013.jpg>]

[HTTP request 1/1]

[Response in frame: 69]

0000	4c 8b ef 1e af 4f a0 48 1c d3 f0 d1 08 00 45 00	L....O.HE.
0010	02 6b 07 f9 40 00 80 06 00 00 c0 a8 01 0b bd f7	.k..@...
0020	a5 99 f7 94 00 50 e2 79 98 06 25 6f e2 94 50 18P.y ..%o..P.
0030	01 03 27 a2 00 00 47 45 54 20 2f 73 63 2f 78 2f	..'...GE T /sc/x/
0040	64 65 66 61 75 6c 74 2f 32 30 31 35 2f 30 39 2f	default/ 2015/09/
0050	30 31 2f 30 30 31 32 31 34 34 31 31 33 32 33 36	01/00121 44113236
0060	33 37 31 33 37 39 39 38 35 37 2f 46 6f 74 6f 2f	37137998 57/Foto/
0070	32 30 31 33 2e 6a 70 67 20 48 54 54 50 2f 31 2e	2013.jpg HTTP/1.
0080	31 0d 0a 48 6f 73 74 3a 20 77 77 77 2e 6c 61 76	1..Host: www.lav
0090	6f 7a 64 65 67 61 6c 69 63 69 61 2e 65 73 0d 0a	ozdegali cia.es..
00a0	43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70	Connecti on: keep
00b0	2d 61 6c 69 76 65 0d 0a 43 61 63 68 65 2d 43 6f	-alive.. Cache-Co

Cuerpo del mensaje

Protocolo HTTP

Ejemplo

Mensaje Response

Ejemplo de una transacción

```
▼ Hypertext Transfer Protocol
> HTTP/1.1 304 Not Modified\r\n
Content-Type: image/jpeg\r\n
Last-Modified: Tue, 01 Sep 2015 18:32:57 GMT\r\n
ETag: "e3d761-7ff5-51eb3c771eff7"\r\n
Cache-Control: max-age=77390\r\n
Date: Wed, 30 Dec 2015 00:03:50 GMT\r\n
Connection: keep-alive\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.030939000 seconds]
\[Request in frame: 62\]
```

Línea inicial

Encabezados

Línea en blanco

```
0000 a0 48 1c d3 f0 d1 4c 8b ef 1e af 4f 08 00 45 00 .H....L. ...O..E.
0010 01 0b d1 8d 40 00 3c 06 47 1b bd f7 a5 99 c0 a8 ....@.<. G.....
0020 01 0b 00 50 f7 94 25 6f e2 94 e2 79 9a 49 50 18 ...P...%o ...y.IP.
0030 02 30 b8 fd 00 00 48 54 54 50 2f 31 2e 31 20 33 .0....HT TP/1.1 3
0040 30 34 20 4e 6f 74 20 4d 6f 64 69 66 69 65 64 0d 04 Not M odified.
0050 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 69 .Content -Type: i
0060 6d 61 67 65 2f 6a 70 65 67 0d 0a 4c 61 73 74 2d mage/jpe g..Last-
0070 4d 6f 64 69 66 69 65 64 3a 20 54 75 65 2c 20 30 Modified : Tue, 0
0080 31 20 53 65 70 20 32 30 31 35 20 31 38 3a 33 32 1 Sep 20 15 18:32
0090 3a 35 37 20 47 4d 54 0d 0a 45 54 61 67 3a 20 22 :57 GMT. .ETag: "
00a0 65 33 64 37 36 31 2d 37 66 66 35 2d 35 31 65 62 e3d761-7 ff5-51eb
00b0 33 63 37 37 31 65 66 66 37 22 0d 0a 43 61 63 68 3c771eff 7"..Cach
00c0 65 2d 43 6f 6e 74 72 6f 6c 3a 20 6d 61 78 2d 61 e-Contro l: max-a
00d0 67 65 3d 37 37 33 39 30 0d 0a 44 61 74 65 3a 20 ge=77390 ..Date:
00e0 57 65 64 2c 20 33 30 20 44 65 63 20 32 30 31 35 Wed, 30 Dec 2015
00f0 20 30 30 3a 30 33 3a 35 30 20 47 4d 54 0d 0a 43 00:03:5 0 GMT..C
0100 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d onnectio n: keep-
0110 61 6c 69 76 65 0d 0a 0d 0a alive... .
```

Cuerpo del mensaje

Práctica 1 Grupal

Análisis del Protocolo HTTP

Práctica 1 Individual

Análisis del Protocolo HTTP