

PROGRAMACIÓN AVANZADA EN JAVASCRIPT

Módulo 3





CLASE 3

Introducción

HTTP

Mensajes HTTP

Modelo Cliente-Servidor

Servidores

Códigos HTTP Response

API

jQuery

Introducción

Entre las clases 3 y 4 del presente Módulo, trabajaremos con varios conceptos que nos permiten trabajar con datos de otras páginas y servidores.

Estos serán nuestros próximos Objetivos:

- Cuáles son los mensajes HTTP.
- Qué es el modelo cliente-servidor.
- Qué significa el "Error 404".
- Qué es una API y cómo se consume.
- Qué es jQuery.
- Qué es AJAX, JSON y XML.

¿Qué es *HTTP*?

Como vimos al principio del curso, **Internet es una red de redes**, conectada por miles de nodos que se comunican entre sí.

Para que la transferencia de información sea posible, es preciso que se cumplan ciertas normas; es decir: **un protocolo**.

El **protocolo usado para transferir información en la web es HTTP** (siglas en inglés de Hypertext Transfer Protocol o, en castellano, Protocolo de Transferencia de hipertexto).

Según Wikipedia, "**HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse**".

Mensajes *HTTP*

Dentro del protocolo HTTP existen 2 tipos de mensajes: por un lado, el pedido que hace el cliente (llamado Request) y, por el otro, la respuesta que envía el servidor (llamado Response).

Requests

Dentro de los Requests existen distintos métodos que indican qué tipo de pedido haremos al servidor. Veamos qué verbos usan estos métodos para especificar lo que solicitan:

Verbo	Descripción
GET	Sirve para obtener datos.
POST	Le dice al servidor que necesita guardar información.
PUT	Indica que se quiere editar información.
DELETE	Sirve para eliminar algo de la base de datos.

Modelo Cliente-Servidor

El **Modelo Cliente - Servidor** refiere a la interacción entre el cliente y el servidor.

Importante: El cliente puede ser uno o varios.

Un cliente puede ser una computadora, un teléfono celular, una tablet, etc. conectada a Internet.

Un servidor, en cambio, es una computadora que almacena las páginas Web y que tiene medios para enviarlas a los clientes.

¿Cómo funciona una Página Web?

A grandes rasgos, una página web funciona de la siguiente manera:

- El cliente usa un navegador (como Chrome, por ejemplo).
- El cliente realiza una petición al servidor a través de internet (por ejemplo, haciendo click en un link).
- El servidor recibe el pedido, elabora la respuesta y se la envía al cliente.
- El navegador del cliente recibe esa respuesta y muestra el contenido en pantalla.



Servidores

Los servidores tienen este nombre porque están al servicio de otras computadoras.

Como gestionan los recursos de una red, deben permanecer siempre encendidos. Si un servidor se apaga, o tiene errores, genera que los usuarios de la red tengan problemas, ya que no podrán disponer de sus servicios.

¿Sabías qué...? En líneas generales, los servidores operan sin ninguna intervención manual y solo deben ser configurados inicialmente.



Códigos para *HTTP Response*

Cuando el servidor responde, además de enviar los datos pedidos, siempre genera un código numérico (de 3 dígitos) que representa el estado de la acción realizada.

El primer dígito de cada código nos da la información necesaria para entender cómo se ejecutó esa acción:

- **1:** Es una petición que devuelve información acerca del servidor.
- **2:** La operación fue satisfactoria.
- **3:** La operación se redirigió a otro sitio (otra dirección URL).
- **4:** Hay un error en el pedido del cliente. Es decir, el Request generado tiene algún error.
- **5:** Hubo un error en el servidor y no se ejecutó satisfactoriamente la acción solicitada (por ejemplo, si el servidor se quedó sin espacio en el disco).

¿Qué significa el famoso "Error 404"?

El *Error 404* es uno de los errores más comunes que ocurren cuando navegamos en internet. Este tipo de error indica que hay un enlace roto, defectuoso o que ya no existe.



404
Not Found

¿Qué son las *API*?

Las API (del inglés Application Programming Interface o, en castellano, Interfaz de Programación de Aplicaciones) son, básicamente, un conjunto de reglas que definen cómo se comunican dos aplicaciones entre sí.

En otras palabras, las **API** son como un **punto que comunica 2 aplicaciones**. Es decir, una API permite el traspaso de información y, a su vez, permite recibir retroalimentación de qué fue lo que sucedió con el requerimiento.

Para usar una API de manera sencilla, utilizaremos la librería **jQuery**.

jQuery

jQuery es una de las librerías más populares de JavaScript. Es decir, es un archivo de JavaScript que tiene métodos y propiedades que podemos usar en nuestro script.

jQuery nos permite seleccionar, manipular y crear elementos; agregar event listeners; crear animaciones y efectos y utilizar otras funcionalidades que vamos a aprender en este módulo.

Tip: Podés ver las más de 9000 líneas de código de jQuery o tener una documentación que te ayude a encontrar qué querés hacer.



¿Cómo agregar *jQuery*?

Al igual que con Bootstrap, hay 2 maneras de agregar jQuery a una página.

- Descargarla para usarla de forma local, usando el atributo src:

```
<script src="jquery-3.5.1.min.js"></script>
```

- Usarla desde un CDN (es decir, una copia online), especificando su ruta con el atributo src:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

Importante: Coloque siempre el script de jQuery antes de su script.

Para saber si agregó correctamente jQuery en su documento, escriba **jQuery en la consola**. Si no le devuelve un error significa que está todo bien :D

MÓDULO 3

+

o

.

GRACIAS

Aquí finaliza la clase n°3 del módulo 3

OTEC PLATAFORMA 5 CHILE