educación IT
CODE YOUR FUTURE

ReactJS Developer

Módulo 6

educación IT
CODE YOUR FUTURE

Introducción a *HTTP*

¿Qué es *HTTP*?

HTTP (*HyperText Transfer Protocol*) es un **protocolo para el intercambio de información en la red,** es decir que **define las reglas de ese intercambio.** HTTP responde al modelo **Cliente/Servidor:** el **cliente** *envía* una **petición** y *recibe* una **respuesta**. El **servidor** *recibe* **una petición** y envía una respuesta asociada. Por defecto, es **el cliente el que inicia la comunicación** y **el servidor el que responde**. Esto se puede invertir gracias a los ***WebSockets***.

Una característica importante de HTTP es que se trata de **un protocolo sin estado.**

¿Qué significa? Significa que, en lo que concierne a las reglas de este protocolo, **toda la información está contenida en el intercambio.** Esto quiere decir que este protocolo no regula lo que sucede *dentro* del servidor ni *dentro* del cliente (no lo regula, le es **indiferente**), sino únicamente **la comunicación**. En otras palabras: **regula las *entradas* y *salidas* entre los agentes, pero no el *proceso interno de cada agente.***

¿Cuál es la importancia del protocolo HTTP en el desarrollo con React?

Siendo HTTP el protocolo por excelencia para el intercambio de datos en la red y React una herramienta para crear interfaces web bajo este protocolo, podemos decir que el protocolo HTTP **son las reglas para el intercambio de información entre React y otros proyectos a través de la web.**

HTTP **son las reglas del juego de la programación web,** sea tanto en el desarrollo del lado del servidor como del lado del cliente.

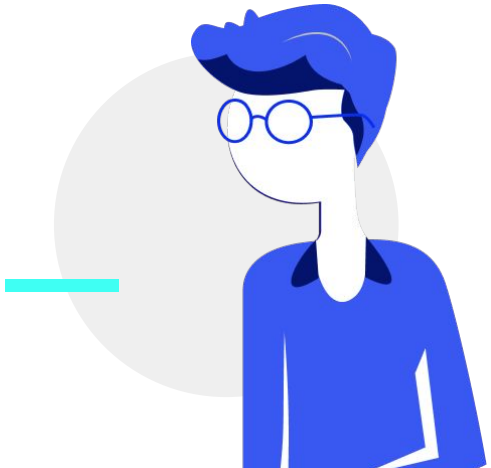
Conocer HTTP para aplicaciones del lado del cliente (las que hacemos con React) nos sirve para definir **una comunicación eficaz con otros sistemas y proyectos desarrollados e implementados fuera de nuestra aplicación.**

¿Qué es un Mensaje HTTP?

Un mensaje, desde HTTP, es la forma genérica de englobar **peticiones** y **respuestas**.

Los mensajes (peticiones y respuestas) de HTTP son, en esencia, **textos**. Es decir, que, en esencia, HTTP **envía y recibe texto formateado.**

- Una **petición** HTTP es **un mensaje enviado del cliente al servidor.**
- Una **respuesta** HTTP es **un mensaje enviado del servidor al cliente.**



¿Cuáles son las partes más importantes de una petición HTTP?

Veamos un ejemplo de un texto de petición HTTP

```
GET /productos HTTP/1.1
Content-type: application/json

{
 “buscar”: “productos”
}
```

En este texto vemos las tres partes principales de una petición HTTP:

- La primera línea contiene la denominada **`Request-Line`.** Es básicamente **el método y la URI.** El método puede ser **cualquier verbo** HTTP como `GET`, `POST`, `DELETE`, etc.
- Debajo de esa línea, van **las cabeceras de la petición.** Una cabecera es un metadato, nos permiten **añadir información extra que deba tener en cuenta el servidor.**
- Por último, se escribe **el cuerpo de la petición.** El cuerpo son **los datos que manda el cliente como información para la petición.**

¿Cuáles son las partes más importantes de una Respuesta HTTP?

Veamos un ejemplo de un texto de Respuesta HTTP:

```
HTTP/1.1 200 OK
Date: Sat, 20 Nov 2021 16:33:35 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1Set-Cookie: 1P_JAR=2021-11-20-16; expires=Mon,
20-Dec-2021 16:33:35 GMT; path=/; domain=.google.com;
Connection: close

<div>Hola mundo</div>
```

Explicación

- En la diapositiva anterior, tenemos la línea inicial que, en el caso de la respuesta, es **la línea de estado.** El Estado HTTP es un número que **resume toda la respuesta.** En este caso, 200 representa que el servidor pudo generar una respuesta de forma satisfactoria.
- Luego, siguen una a una **las cabeceras HTTP,** con la misma lógica que la petición.
- Finalmente, hay **un texto representando el contenido de la respuesta.** Técnicamente, el texto puede ser, justamente, cualquier texto. Por esta razón existe HTML: **es un lenguaje para representar el cuerpo de respuestas HTTP.**

¿Cómo trabajamos con estos datos en PHP?

Es importante recalcar que **PHP nos abstrae de tener que escribir estos textos manualmente.**

Tenemos funciones y herramientas que nos permiten leer la petición entrante y responder al cliente. Sin embargo, y debajo de todas esas herramientas, la entrada a un programa PHP es una petición HTTP y la salida es una respuesta HTTP.

Links de interés

- **Especificación de HTTP**
  RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1
- **Métodos de Petición HTTP**
  Métodos de petición HTTP
- **Tipos de Cuerpo disponibles**
  Lista completa de tipos MIME - HTTP | MDN
- **Códigos de Estado HTTP**
  Códigos de estado de respuesta HTTP
- **Cabeceras HTTP disponibles**
  HTTP headers

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!