

## Experiencia con la instalación de la tecnología

Debido a que esta tecnología se utiliza en el lado del cliente, la instalación de la misma consistía en importar jQuery a través de un CDN, tal como se muestra a continuación:

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

Como mencionan los compañeros, al utilizar jQuery se pueden acceder a los elementos de la página de una manera más sencilla y a la vez incluye AJAX para realizar las peticiones al servidor.

Además, para efectos del taller fue necesario instalar Node.js para correr un servidor local y hacer las peticiones al mismo. Esto se realizó de manera sencilla con el instructivo que los compañeros proporcionaron.

## Aspectos más relevantes de AJAX

Según lo que comentan los compañeros, AJAX (**A**synchronous **J**avaScript and **X**ML) es una tecnología que surge alrededor de 2005 debido a la necesidad de crear páginas web dinámicas, su creador fue Jesse James Garret. Además, es una tecnología utilizada para crear aplicaciones que son ejecutadas en el navegador y que cuentan con una conexión en segundo plano con el servidor. Esto con el fin de obtener datos y mostrarlos en la página según la necesidad de los usuarios y sin tener que recargar toda la página web cuando ocurre un cambio, evitando interferir con la usabilidad de la misma.

Además, los compañeros mencionan que para realizar las peticiones al servidor se debe incluir un objeto XMLHttpRequest en el archivo JavaScript de la página web. También, dicho objeto es el que se encarga de realizar las peticiones al servidor y recibir los datos para luego mostrarlos en la página web. Asimismo, los compañeros mencionan que el objeto XMLHttpRequest tiene algunas funciones principales:

- a. **onreadystatechange:** se ejecuta cada vez que el estado del objeto XMLHttpRequest cambia.
- b. **readyState:** tiene valores de 0 a 4 que indican el estado actual del objeto XMLHttpRequest.
- c. **status:** contiene el estado de la petición cuando finaliza, es decir si fue exitosa o no.
- d. **responseText:** contiene los datos proporcionados por el servidor al hacer el request.

Por consiguiente, entre las principales ventajas que los compañeros mencionan, se encuentran:

- Incremento de la velocidad, usabilidad e interactividad con el usuario al no tener que recargar toda la página cuando ocurre un cambio.
- Soportado por varios navegadores como Mozilla Firefox, Google Chrome, Microsoft Edge, entre otros.

Por otro lado, entre las principales desventajas que los compañeros mencionan, se encuentran:

- Problemas de acceso cuando se hacen peticiones con muchos parámetros.
- El URL no se actualiza y si se requieren los parámetros de nuevo, se deben volver a pedir.
- Algunos navegadores o versiones de los mismos no soportan JavaScript.

## Descripción del taller

El taller realizado por los compañeros consistía en utilizar el método **load** de jQuery, con el fin de obtener información acerca de un personaje. Dicha información se obtiene de un archivo de texto plano localizado en el servidor.

Además, para efectos del taller se crearon tres archivos cuyo contenido se muestra a continuación:

### a. index.html

```
index.html > @ html
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset='utf-8'>
6      <meta http-equiv='X-UA-Compatible' content='IE=edge'>
7      <title>AJAX Example</title>
8      <meta name='viewport' content='width=device-width, initial-scale=1'>
9      <link rel='stylesheet' type='text/css' media='screen' href='index.css'>
10     <script src='https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js'></script>
11     <script src='index.js'></script>
12
13 </head>
14
15 <body>
16     <div class='center'>
17         <img src='https://upload.wikimedia.org/wikipedia/commons/thumb/8/82/CharlesBabbage.jpg/220px-CharlesBabbage.jpg'.alt='Charles Babbage'>
18         <div>
19             <button id='request'>Learn more about Babbage</button>
20             <div id='bio'></div>
21         </div>
22     </div>
23     <script type='text/javascript' src='./index.js'></script>
24 </body>
25
26 </html>
```

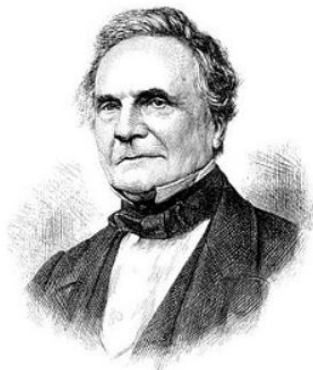
## b. index.js

```
JS index.js > ...
1  $btn = $('#request');
2  $bio = $('#bio');
3
4  $btn.on('click', function() {
5      $(this).hide();
6      $bio.load('bio.txt', completeFunction);
7  });
8
9  function completeFunction(responseText, textStatus, request) {
10     $bio.css('border', '1px solid #e8e8e8');
11     if (textStatus == 'error') {
12         $bio.text('An error occurred during your request: ' + request.status + ' ' + request.statusText);
13     }
14 }
```

## c. index.css

```
1  .center {
2      text-align: center;
3  }
4
5  button {
6      margin: 0 auto;
7  }
```

Finalmente, el resultado que se obtuvo al realizar el taller fue el siguiente:



[Learn more about Babbage](#)



Considered by some to be a "father of the computer", Babbage is credited with inventing the first mechanical computer that eventually led to more complex electronic designs, though all the essential ideas of modern computers are to be found in Babbage's Analytical Engine. His varied work in other fields has led him to be described as "pre-eminent" among the many polymaths of his century. Parts of Babbage's incomplete mechanisms are on display in the Science Museum in London. In 1991, a functioning difference engine was constructed from Babbage's original plans. Built to tolerances achievable in the 19th century, the success of the finished engine indicated that Babbage's machine would have worked.