

**Apunte Ajax**  
**Introducción al Desarrollo de Páginas WEB**  
**Rodrigo Isaac Venegas Vega**  
**2017047627**

**Instalación:**

El proceso de instalación fue muy sencillo, no hubo ningún problema. La instalación de NoseJS es sencilla al igual que la instalación del paquete http-server. Sin embargo, perdí un poco de tiempo a la hora de iniciar el servidor ya que no lo inicié en la carpeta de mi trabajo y eso me causó un problema.

**Aspectos relevantes:**

- Asynchronous JavaScript And XML
- Después de la exposición de los compañeros, logré entender mejor la finalidad de Ajax y aprendí lo siguiente:
- Creado en 2005, debido a la necesidad de apps WEB dinámicas que pudiesen actualizar información sin necesidad de recargar la página.
- Es posible realizar cambios en la página sin necesidad de hacer recarga. Esto incrementa la escalabilidad, usabilidad, interacción con el usuario y experiencia de usuario.
- Conexión asíncrona. Redirecciona datos al servidor en segundo plano sin interferir en el funcionamiento de la página.
- Sus funciones de llamada se ejecutan en JS y los accesos a los datos se realiza mediante XML HttpRequest.
- Entre las tecnologías que utiliza, incluye HTML, Css, DocumentObjectModel, XML HttpRequest. Los datos pueden almacenarse en txt, html, en otro servidor, etc.
- Es multiplataforma, compatible con Chrome, FireFox, Safari, Opera.
- Una desventaja es que si queremos recuperar datos, es necesario volver a pedirlos al usuario. No es posible llevar un historial de los procesos avanzados por el usuario.

**Descripción del taller:**

Para un mayor entendimiento de AJAX, se realizó un pequeño taller guiado por los integrantes del equipo AJAX. El objetivo de este taller es crear una página con un servidor que muestre una imagen de algún personaje famoso de nuestra preferencia y, al presionar un botón, debería desplegar información de este personaje.

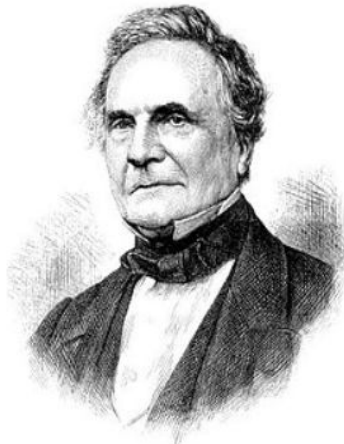
1. Forma final de la página:

169.254.191.216:8080/



[Learn more about Babbage](#)

2. Al presionar el botón:



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.

3. Se utiliza un único html con instrucciones muy simples:

```
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         
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>
```

4. Para dar formato a lo creado en el html, se implementa un poco de código en Css:

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

}

5. Además, se hace uso de un archivo JS para el back-end, se le da funcionalidad al botón que despliega la información:

```
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 }
```

6. Para almacenar el texto de información de hace uso de un TXT 'bio'.