

Ajax

pedromanriquez

Published
with GitBook



Tabla de contenido

Introduction	0
¿Cómo usar AJAX?	1
Flujo del software con AJAX	2
Ejemplo	3
Actividad	4



Introducción a AJAX

AJAX significa Asynchronous JavaScript And XML (Javascript Asíncrono y XML), Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

Para el uso de AJAX utilizaremos la biblioteca de Javascript llamada "jQuery", así simplificaremos un poco el uso de esta tecnología.

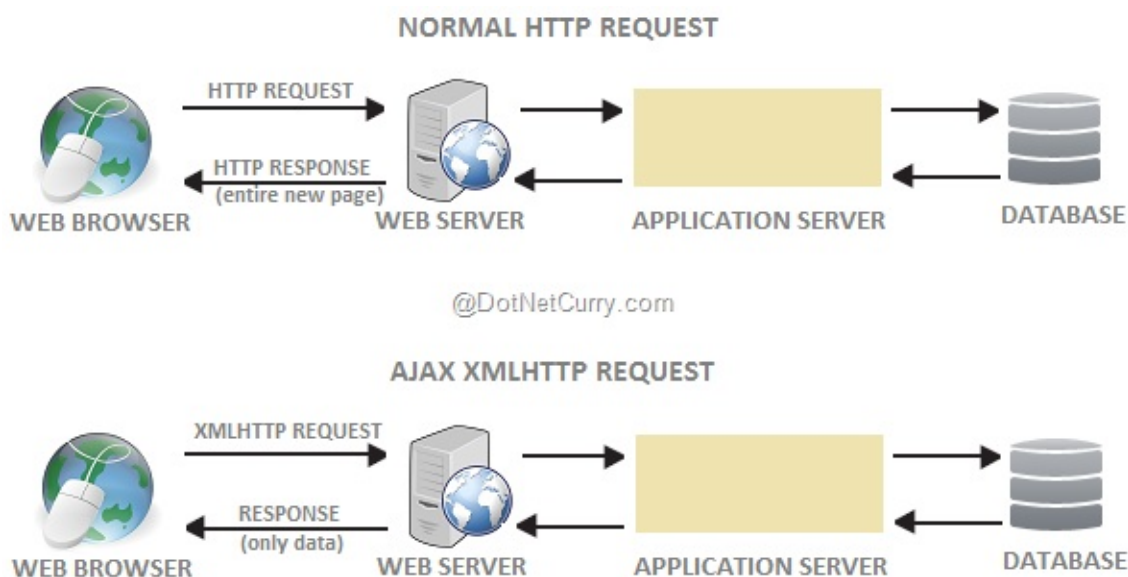
Además realizaremos un ejemplo utilizando una base de datos pequeña de solo dos tablas, una de personas y otra de cargos. De esta forma podremos hacer peticiones al servidor para poder hacer consultas a la base de datos y podremos agregar nuevos elementos a nuestro HTML sin la necesidad de salir del sitio y de esta forma agregar elementos de forma dinámica.

¿Cómo usar AJAX en mi página web?

Bueno primero debemos definir AJAX, como decíamos en la introducción AJAX es una tecnología que nos permitirá realizar una petición a un servidor y sin la necesidad que redireccionar a otra página, esperamos la respuesta y añadimos elementos a nuestra vista, es por eso que es de tipo asíncrono y esto hace que nuestra web sea dinámica.

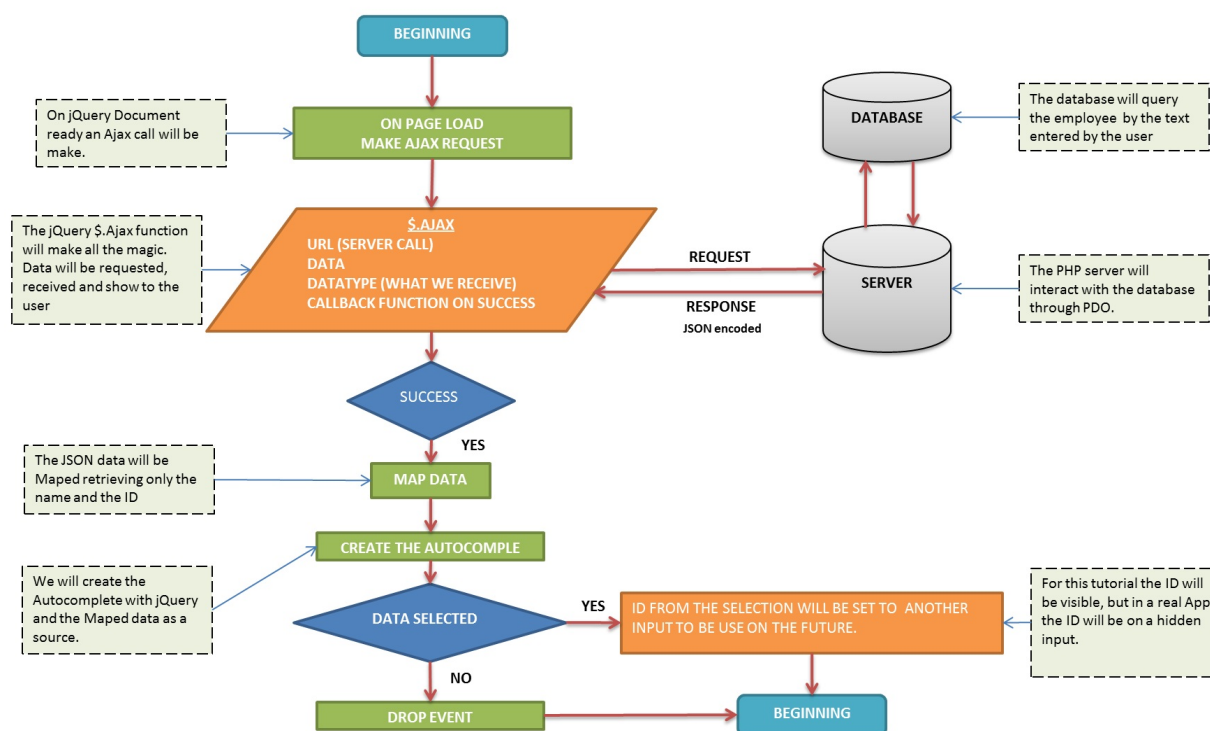
Como mencionamos anteriormente el proceso es que nuestro archivos javascript deben realizar una petición a un archivo del servidor (en este caso php) ese archivo puede cargar una lista o hacer una consulta a una base de datos, esperan la respuesta y la procesan en una función llamada denominada "callback" que procesará la respuesta del servidor para agregar el contenido que necesitemos en nuestro HTML.

La siguiente imagen ilustra la petición de una página web sin AJAX y con AJAX, la diferencia está en la respuesta del servidor.



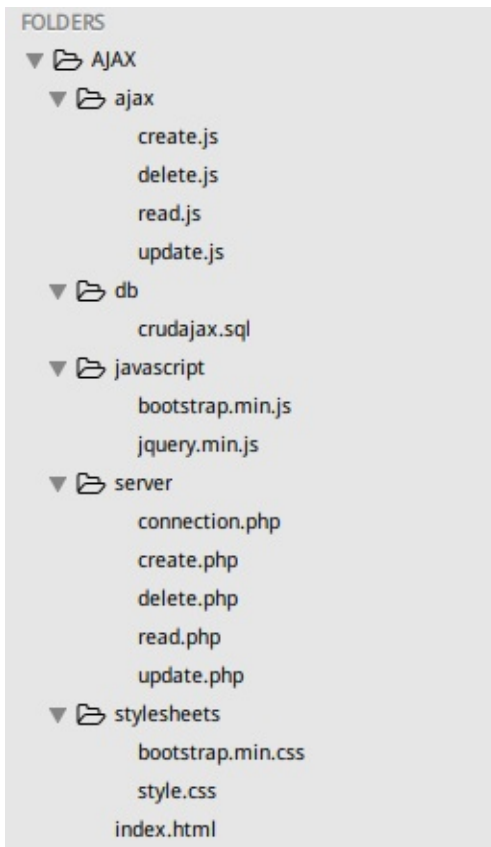
Flujo del software

En la siguiente imagen se trata de explicar aún más a fondo como funciona una petición desde un archivo de javascript a un servidor y como después este procesa la información de respuesta del servidor.



Ejemplo:

En el siguiente código crearemos un proyecto y ordenaremos los archivos con una estructura de carpetas para separar en cierta forma las distintas funciones dentro del proyecto, la siguiente imagen ilustrará lo mencionado anteriormente:



el código del archivo index.html es el siguiente:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>AJAX JQuery</title>
  <link rel="stylesheet" href="stylesheets/bootstrap.min.css">
  <link rel="stylesheet" href="stylesheets/style.css">
  <script src="javascript/jquery.min.js"></script>
  <script src="javascript/bootstrap.min.js"></script>
</head>
<body>
  <div class="container">
    <div class="well">
      <button class="btn btn-success btn-lg" data-toggle="modal" href='#agregar'>Agregar
      <table class="table table-condensed table-hover">
        <thead>
```

```

        <tr class="info">
            <th>RUT</th>
            <th>NOMBRE</th>
            <th>EMAIL</th>
            <th>TELÉFONO</th>
            <th>CARGO</th>
            <th>ACCIÓN</th>
        </tr>
    </thead>
    <tbody id="tabla-registro">
        <!-- contenido dinámico -->
    </tbody>
</table>
</div>
</div>
<!-- Modal ingreso persona -->
<div class="modal fade" id="agregar">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;
            </div>
            <div class="modal-body">
                <div class="modal-body">
                    <form id="form-ingreso">
                        <legend>Nuevo Registro:</legend>
                        <div class="form-group">
                            <input required type="text" class="form-control" placeholder="Ingrese el RU
                        </div>
                        <div class="form-group">
                            <input required type="text" class="form-control" placeholder="Ingrese Nomb
                        </div>
                        <div class="form-group">
                            <input required type="email" class="form-control" placeholder="Ingrese Corr
                        </div>
                        <div class="form-group">
                            <input required type="text" class="form-control" placeholder="Ingrese Telef
                        </div>
                        <div class="form-group">
                            <select required name="cargo" class="form-control btn btn-default">
                                <option value="12">Jefe de Proyecto</option>
                                <option value="23">Analista</option>
                                <option value="32">Desarrollador</option>
                            </select>
                        </div>
                        <div class="form-group">
                            <input type="submit" class="btn btn-success btn-lg" value="Ingresar">
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
</div>

```

```
<!-- Fin Modal ingreso persona -->
<script src="ajax/create.js"></script>
<script src="ajax/delete.js"></script>
<script src="ajax/read.js"></script>
<script src="ajax/update.js"></script>
</body>
</html>
```

Se puede ver que incorporamos Bootstrap en nuestro proyecto y además tendremos varios archivos JavaScript y estos son los que usarán la técnica AJAX, así se verá el archivo descrito en conjunto con sus modales.

Agregar Persona

RUT	NOMBRE	EMAIL	TELÉFONO	CARGO	ACCIÓN
-----	--------	-------	----------	-------	--------

Modal de ingreso:

Formulario para ingresar persona

Nuevo Registro:

Ingrese el RUT

Ingrese Nombre con Apellido

Ingrese Correo Electrónico

Ingrese Telefono Móvil

Jefe de Proyecto

Ingresar

Teniendo en cuenta que el ejercicio es hacer un CRUD (Create Read Update Delete) mediante el uso de AJAX ahora estudiaremos el archivo "ajax/create.js" y el "server/create.php", el archivo JavaScript usará JQuery para poder hacer peticiones AJAX a "server/create.php" y esperar la respuesta para agregar elementos al HTML, de la siguiente forma:


```
$('#form-ingreso').submit(function(event){
    event.preventDefault();

    $.post('../server/create.php',
        $('#form-ingreso').serialize(),
        function(dato){
            $("#agregar").modal('hide');
            if(dato.exito){
                console.log(dato);
                var variable="";
                if(dato.cargo == 12) variable="Jefe Proyecto";
                if(dato.cargo == 23) variable="Analista";
                if(dato.cargo == 32) variable="Desarrollador";

                $('#tabla-registro').after(
                    '<tr id="fila_'+dato.rut+'">'+
                    '<td>'+dato.rut+'</td>'+
                    '<td>'+dato.nombre+'</td>'+
                    '<td>'+dato.email+'</td>'+
                    '<td>'+dato.telefono+'</td>'+
                    '<td>'+variable+'</td>'+
                    '<td> <button value="'+dato.rut+'" type="button" class="btn
                    '</tr>'
                );
            }else{
                alert('Error al recibir la respuesta del Servidor');
            }
        });
});
```

Para poder entender este código también debemos entender el "server/create.php", puesto que están relacionados directamente por la petición del archivo "ajax/create.js", el código del lado del servidor es el siguiente:

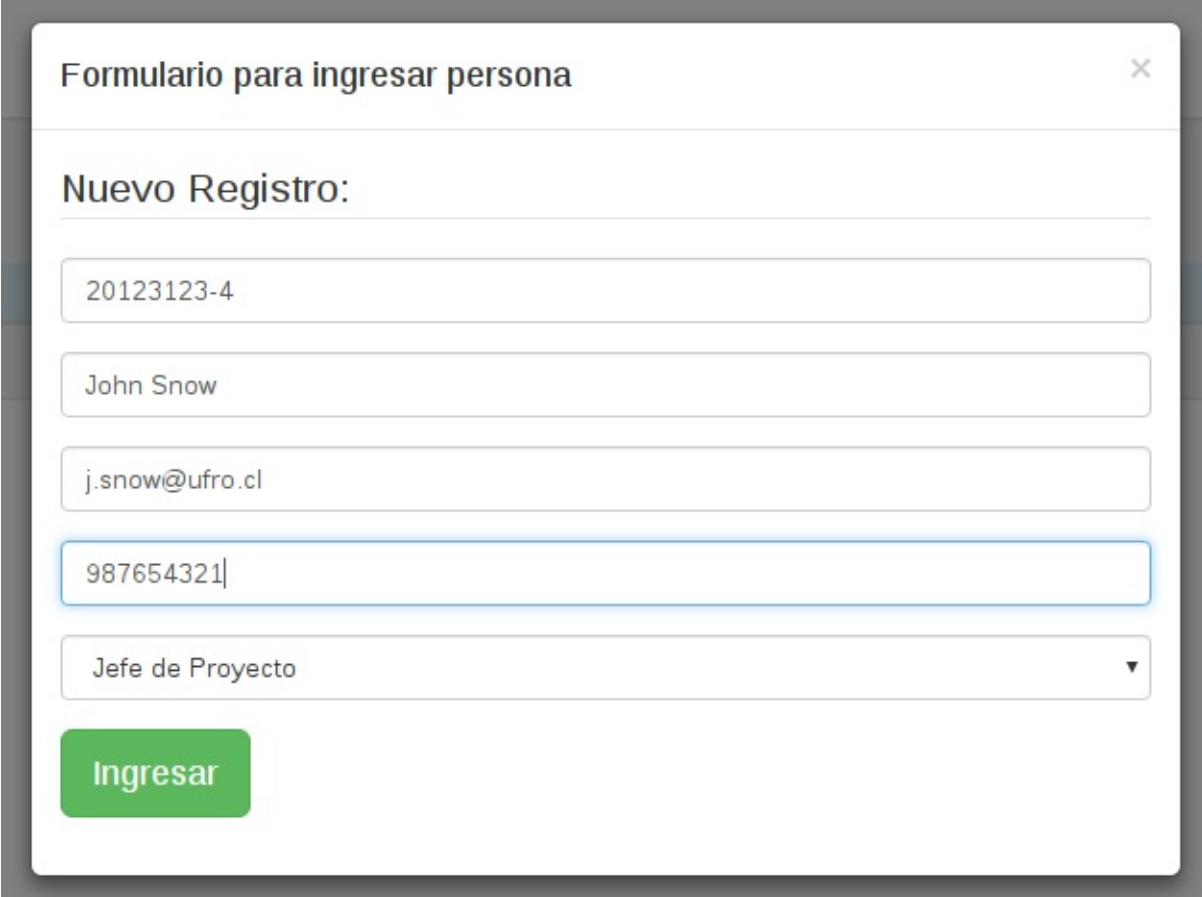
```
<?php
require('connection.php');
$db= new ConnectionDB();

$rut = $db->escape_string($_POST['rut']);
$nombre = $db->escape_string($_POST['nombre']);
$mail = $db->escape_string($_POST['email']);
$telefono = $db->escape_string($_POST['telefono']);
$cargo = $db->escape_string($_POST['cargo']);

/* Enviamos la instrucción SQL que permite ingresar
los datos a la BD en la tabla contactos */
if($db->query("insert into contacto values ('$rut','$nombre','$mail','$telefono', '$car
header('Content-Type: application/json');
echo json_encode(array('exito'=>true, 'rut'=>$rut, 'nombre'=>$nombre, 'email'=>$mail, '
}else{
die("Ocurrio UN problema al ejecutar la consulta de insercion en BBDD error [ ".$db->
}

?>
```

Después de tener estos códigos funcionando podremos agregar un nuevo registro en nuestro sistema de la siguiente forma:



Formulario para ingresar persona

Nuevo Registro:

20123123-4

John Snow

j.snow@ufro.cl

987654321

Jefe de Proyecto

Ingresar

Dejando el siguiente resultado:

Agregar Persona

RUT	NOMBRE	EMAIL	TELÉFONO	CARGO	ACCIÓN
20123123-4	John Snow	j.snow@ufro.cl	987654321	Jefe Proyecto	<div>Actualizar</div> <div>Eliminar</div>

Toda las instrucciones necesarias para ejecutar el proyecto se encuentran en la actividad de la guía, puesto que falta mencionar el archivo de conexión a la base de datos.

Actividad

Luego de estudiar el ejemplo y haber terminado la lectura de la guía, se pedirá que clone en su equipo el siguiente enlace:

<https://github.com/PedroManriquez/ActividadAjax.git>

Una vez clonado el repositorio usted deberá crear una base de datos e importar el archivo sql ubicado en "db/crudajax.sql", luego tendrá una base de datos con dos tablas y algunos registros, el siguiente paso será modificar el archivo de la configuración de la base de datos "server/connection.php", y cambiar la contraseña de su usuario root de phpmyadmin, con esto ya está en condiciones de ejecutar el proyecto y de realizar las siguientes funcionalidades dentro del proyecto.

- Actualmente el archivo "ajax/read.js" cuenta con un "get" de JQuery que hace una llamada a "server/read.js" (retorna todas las filas de una consulta en sql), pero el archivo javascript **NO** agrega elementos en el index.html, por ende, Usted deberá escribir el código necesario para agregar la lista de personas en la tabla del archivo html (puede basarse en el archivo "ajax/create.js" puesto que la persona que agrega en la BBDD también lo agrega en la tabla del index.html).

[Imagen de evidencia de retorno "server/read.php"]

▼ Object ⓘ	read.js:5
cargo: "Jefe Proyecto"	
email: "j.snow@ufro.cl"	
nombre: "John Snow"	
rut: "20123123-4"	
telefono: "987654321"	
▶ __proto__: Object	
▼ Object ⓘ	read.js:5
cargo: "Analista"	
email: "a.guzman@gmail.cl"	
nombre: "Andres Guzman"	
rut: "12234432-4"	
telefono: "78787878"	
▶ __proto__: Object	
▼ Object ⓘ	read.js:5
cargo: "Analista"	
email: "r.rodriguez02@ufromail.cl"	
nombre: "Rodrigo Rodriguez"	
rut: "18234567-8"	
telefono: "23232323"	
▶ __proto__: Object	
▼ Object ⓘ	read.js:5
cargo: "Analista"	
email: "ptrmanriquez@gmail.com"	
nombre: "Pedro Manriquez"	
rut: "18683667-7"	
telefono: "84823178"	
▶ __proto__: Object	

- Realizar la funcionalidad del boton eliminar ubicado en cada registro, este deberá consultar si realmente esta seguro de borrar el registro y debe borrar la fila en la base de datos y en la tabla del archivo index.html.
- Realizar la funcionalidad del boton actualizar o editar ubicado en cada registro, usted deberá implementar un segundo modal para actualizar un contacto (este debe ser un formulario como el de ingreso, pero debe contener en los inputs el valor actual del registro seleccionado) también el campo de RUT debe estar deshabilitado, el cambio debe reflejarse tanto en la base de datos como en la tabla del archivo index.html.