

1 Ejemplo paso a paso de una aplicación que usa AJAX

En este otro tutorial práctico vamos a analizar un poco más en detalle y poco a poco la tecnología Ajax.

?La simplicidad de lo Nuevo.?

Lo primero que necesitamos es crear una nueva variable y asignarla a una instancia del objeto **XMLHttpRequest**.

Es bastante sencillo en Javascript; simplemente tenemos que usar el comando `new` con el nombre del objeto tal y como aparece en el listado 1.

1.1 Listado 1. Crear un Nuevo objeto XMLHttpRequest

```
<script language="javascript">
  var request = new XMLHttpRequest();
</script>
```

Bueno, pues ya hemos creado en Javascript una variable llamada `request` y le hemos asignado una nueva instancia del objeto `XMLHttpRequest`.

A partir de este momento ya podemos comenzar a usar las funciones de ese objeto.

1.2 Control de errores

En la vida real esta creación del objeto `XMLHttpRequest` puede que no funcione correctamente, ya que puede darse el caso de que muchos navegadores antiguos no soporten este objeto.

Por lo tanto necesitaremos comprobar si el objeto se ha podido crear correctamente o no.

El listado 2 muestra como se puede gestionar si se ha producido algún error en la creación de dicho objeto.

1.3 Listado 2. Crear un objeto XMLHttpRequest con control de errores en la creación.

```
<script language="javascript">

var request = false;
try {
  request = new XMLHttpRequest();
}
catch (failed)
{
  request = false;
}

if (!request)
  alert("Error inicializando XMLHttpRequest!");
</script>
```

Vamos a explicar un poco más a fondo estos pasos:

1. Creamos una nueva variable llamada `request` y le asignamos un valor `false`. Usaremos la condición `false` para indicar que el objeto `XMLHttpRequest` no ha podido ser creado.
2. Añadimos un bloque `try/catch` para control de errores:
3. Dentro de `Try` intentará crear el objeto `XMLHttpRequest`.
4. Si ocurre algún error (`catch (failed)`), nos aseguramos de asignar a la variable `request` un valor `false`.
5. Comprobaremos si la variable `request` es todavía `false` (si la creación del objeto ha ido bien no tendrá el valor `false`, sino que tendrá la referencia al nuevo objeto `XMLHttpRequest` recién creado).
6. Si ha habido algún problema y (`request` es falso) usaremos Javascript para alertar al usuario de que ha habido un problema.

A partir de este momento tenemos un código que creará un objeto `XMLHttpRequest` e incluso nos indicará si el objeto ha podido ser creado

correctamente o no.