respuesta" de http

Decíamos anteriormente que cuando el servidor ha terminado la solicitud, automáticamente llama a la función programada en la propiedad **onreadystatechange** del objeto XMLHttpRequest.

Ésto es cierto pero no es toda la verdad. De hecho llama a la función programada en **onreadystatechange** cada vez que se modifica el estado en el ciclo de solicitud/respuesta.

Sumario

Veamos primero los estados http.

Un estado http indica el estado de una solicitud.

- Se emplea para saber si una solicitud ha sido comenzada, está siendo respondida o si ha sido completada.
- También es útil para conocer cuando podemos leer de forma segura la respuesta en forma de texto o datos que un servidor nos puede estar proporcionando.

Necesitamos conocer sobre los **5 estados en nuestras aplicaciones Ajax**:

0: La solicitud no ha está inicializada (justo antes de haber llamado a open()).
1: La solicitud está inicializada, pero no ha sido enviada (antes de llamar al método send()).
2: La solicitud fue enviada y está siendo procesada (en este punto podremos obtener las cabeceras de respuesta).
3: La respuesta está siendo procesada; a menudo podemos obtener datos parciales de la respuesta, pero el servidor no ha terminado todavía la ejecución.
4: La respuesta está lista; podemos leer la respuesta del servidor y utilizarla.

Dependiendo del navegador podremos obtener múltiples estados. Algunos nunca reportan el estado 0 o 1, y simplemente van directamente al estado 2, 3 y 4. Otros reportan todos los estados, etc.

Para la programación con Ajax el único estado con el que vamos a tratar es el estado 4 (la respuesta está lista), ya que aquí ya podremos obtener la respuesta y trabajar con ella. Dicho estado de la ejecución lo podremos comprobar en la propiedad **readyState** del objeto XMLHttpRequest.

Véase listado 13.

Listado 13. Chequeando el estado de ?respuesta está lista?.

```
function actualizarPagina() {
if (request.readyState == 4)
            alert("El servidor ha terminado la consulta en la pagina php.!");
   }
```

Este cambio se asegura de que el servidor ha terminado realmente el proceso.

Si ejecutamos este proceso veremos que puede funcionar correctamente y en realidad lo hace, pero ¿qué ocurre si nuestra página php devuelve un error, o la página a la que llamamos no se encuentra o no está disponible?

Pues para comprobar esos aspectos externos de la ejecución disponemos de los típicos códigos de error http.

Por ejemplo el **código 404** para indicar que no se encuentra la página, **403** y **401** para indicar que se prohibe el acceso a la página, etc..

Deberemos por lo tanto añadir a nuestra comprobación el estado http. Para ver que todo es correcto tendremos que comprobar el estado con código

200. Este estado http lo comprobaremos consultado la propiedad status de nuestro objeto XMLHttpRequest.

Véase el listado 14, dónde hemos añadido la comprobación del estado http.

Listado 14. Chequeando el estado HTTP

```
function actualizarPagina() {
 if (request.readyState == 4)
      if (request.status == 200)
         alert("El servidor ha terminado la consulta en la pagina php.!");
   }
```

Una version más completa de la comprobación la tenemos en el listado 15.

Listado 15. Añadiendo más comprobación de errores.

```
function actualizarPagina() {
     if (request.readyState == 4)
       if (request.status == 200)
             alert("El servidor ha terminado ejecución OK!");
       else if (request.status == 404)
         alert("La URL no existe!");
       else
         alert("Error: Código " + request.status);
   }
```

Si modificamos la URL en la función obtenerInfoCliente(), por una incorrecta y ejecutamos veremos que nos muestra la alerta de error.