Enviando petición e indicando método de respuesta

Una vez configurada la solicitud con open(), estaremos preparados para enviar la solicitud.

El método empleando para enviar la petición se denomina send().

- Send() necesita un único parámetro: el contenido a enviar.

En este caso el contenido a enviar formará parte de la URL ya que estamos enviando datos con el método GET:

```
var url = "buscarcliente.php?phone=" + escape(phone);
```

Aunque podemos enviar datos empleando el método send(), podemos enviar también datos a través de la URL en sí misma.

De hecho con el método GET (que es el 80% de las solicitudes bajo Ajax), es mucho más fácil enviar datos.

Si vamos a enviar datos de forma segura o XML, entonces tendremos que indicar dicho contenido a enviar dentro del método send().

Si no vamos a enviar datos con el método send(), indicaremos como parámetro null.

Veamos el listado 9 en el cuál enviamos dicho contenido.

Listado 9. Enviando la solicitud.

```
function obtenerInfoCliente() {
     var phone = document.getElementById("phone").value;
     var url = "buscarcliente.php?phone=" + escape(phone);
     request.open("GET", url, true);
     request.send(null);
   }
```

Indicando el método de respuesta.

Disponemos de una **propiedad** en el objeto **XMLHttpRequest** denominada **onreadystatechange**.

Repasemos el listado 9:

Se ha abierto una solicitud y se ha enviado. Debido a que es una solicitud asíncrona, el método Javascript obtenerInfoCliente no esperará por una respuesta para continuar. Por lo que el código continuará hasta el final y devolverá el control al formulario.

Esto crea una interesante pregunta: **¿qué ocurre cuando el servidor ha procesado la solicitud?**

La respuesta, tal y como está el código actualmente es ¡ nada ! Obviamente, esto no es bueno, es decir el servidor necesitará algún tipo de instrucción que le indique qué hacer cuando haya terminado de procesar la petición (hecha por nuestro objeto XMHLHttpRequest a nuestra página php).

Aquí es el momento en el que la propiedad **onreadystatechange** del objeto XMLHttpRequest entra en juego.

Esta propiedad nos permite especificar un método de respuesta. Esa respuesta permite al servidor devolver un código a nuestro script.

Cuando el servidor termina la ejecución de la página php, examina el objeto XMLHttpRequest y específicamente en la propiedad onreadystatechange.

Esa respuesta es inicializada por el servidor, ya que éste realiza una llamada hacia la página web original del cliente, y lo hará en el momento que haya terminado su solicitud.

Aquí es realmente dónde la asincronía entra en juego: el cliente trabaja con el formulario en un nivel, mientras que en otro el servidor responde a una solicitud y ejecuta el procedimiento empleado en la propiedad onreadystatechange.

Por lo tanto, n**ecesitamos especificar un método a ejecutar en dicha propiedad onreadystatechange y que se ejecutará cuando el proceso de la página php haya terminado**.

Véase el listado 10.

Listado 10. Estableciendo el método de respuesta.

```
function obtenerInfoCliente() {
     var phone = document.getElementById("phone").value;
     var url = "buscarcliente.php?phone=" + escape(phone);
     request.open("GET", url, true);
     request.onreadystatechange = actualizarPagina;
     request.send(null);
   }
```

Por favor presta atención a que dicha propiedad está situada justo antes del método send().

Tendremos que especificar dicha propiedad antes de que la solicitud sea enviada, así el servidor podrá actualizar dicha propiedad justo cuando termine la solicitud.