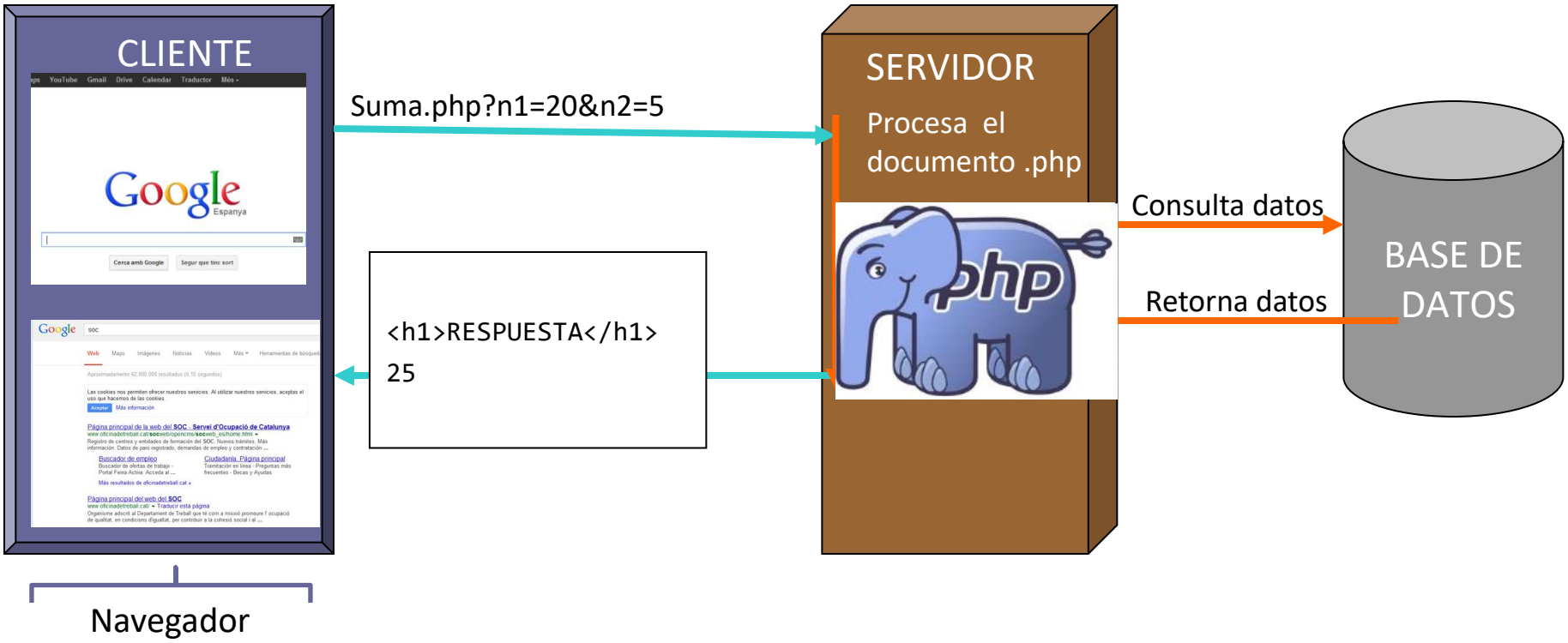
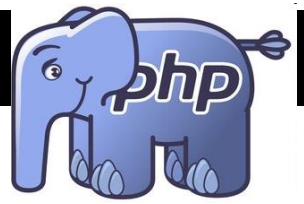
Linkia FP

DAW	M06	T06	Material	AJAX y PHP

SOBRE PHP


CLIENTE
Suma.php?n1=20&n2=5
SERVIDOR
Procesa el documento .php
Consulta datos
Retorna datos
BASE DE DATOS
<h1>RESPUESTA</h1>
25
Navegador

SOBRE PHP



- Se escribe en un documento de texto llano con extensión *.php*
- Puede compartir documento con otros lenguajes´.
- Cada boque de código ha de empezar por **<?php** y terminar con **?>**
- Se ejecuta en el servidor y permite generar datos según la petición del cliente.
- Los datos se imprimen con la función “echo”.
- El cliente puede pasar parametros por GET, POST y otros métodos.
- Las variables se declaran con **$**.
- No se indica el tipo de variable.
- Se concadena con el punto “.”

```
<?php
    $nombre="Eli";
    echo "Helllo".$nombre;
?>
```

PHP y variables de sesión

OBJETIVO

•Iniciar una variable sesión en PHP permite que el servidor reconozca que dos peticiones realizadas en períodos de tiempo distintos son realizadas por el mismo usuario.

•Las variables de sesión nos permiten almacenar información entre dos peticiones realizadas por el mismo usuario.

```
session_start();
if (!array_key_exists("nombre_usu", $_SESSION)) {
     $_SESSION[nombre_usu']='anonimo';
}
echo 'hola'.$_SESSION['nombre_usu'];
```

•El ejemplo anterior guarda dentro la variable `'nombre_usu'` el nombre `'anonimo'` si no existía, y a continuación muestra el nombre del usuario.

Asynchronous JavaScript And XML

•Ajax es una técnica de desarrollo web que permite enviar consultas y recibir respuestas del servidor sin tener que volver a cargar toda la web.

•Para enviar y recibir las consultas se utiliza el objeto JavaScript XMLHttpRequest.

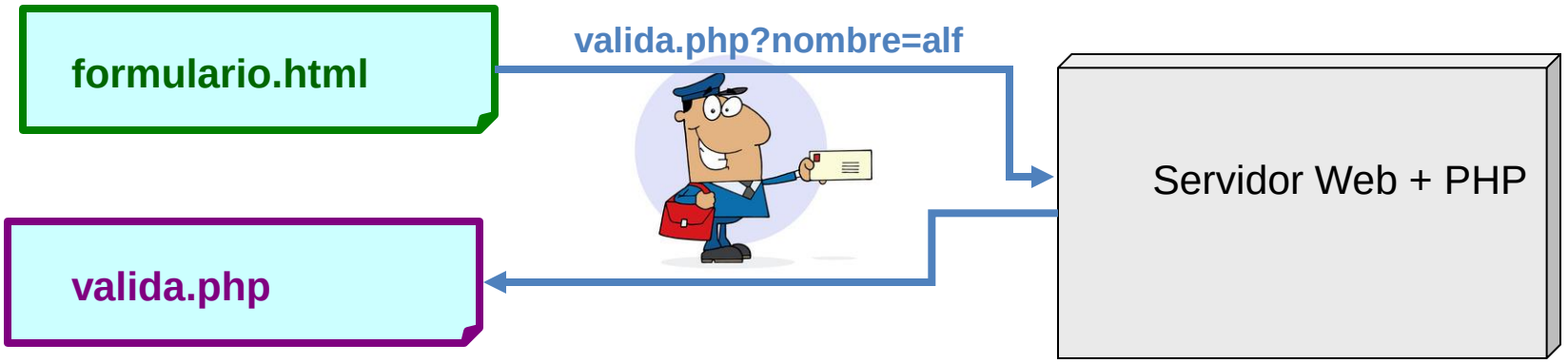
•Originalmente las respuestas del servidor se enviaban codificadas en formato XML. Actualmente se suele utilizar más la codificación JSON.

•Mediante JavaScript deberemos extraer los datos *devueltos por el servidor y mostrárselos al usuario (o realizar las acciones pertinentes).*

Paradigmas comunicación

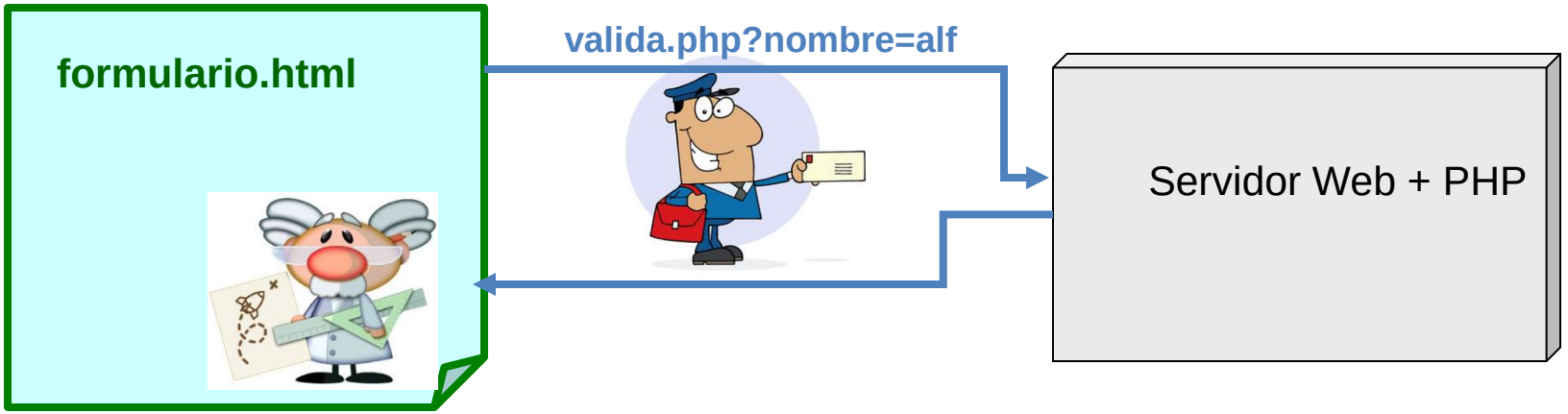
Sin Ajax

Si queremos pedir nueva información a otro sitio, hasta ahora enviábamos la petición, y recibíamos una nueva página con toda la información

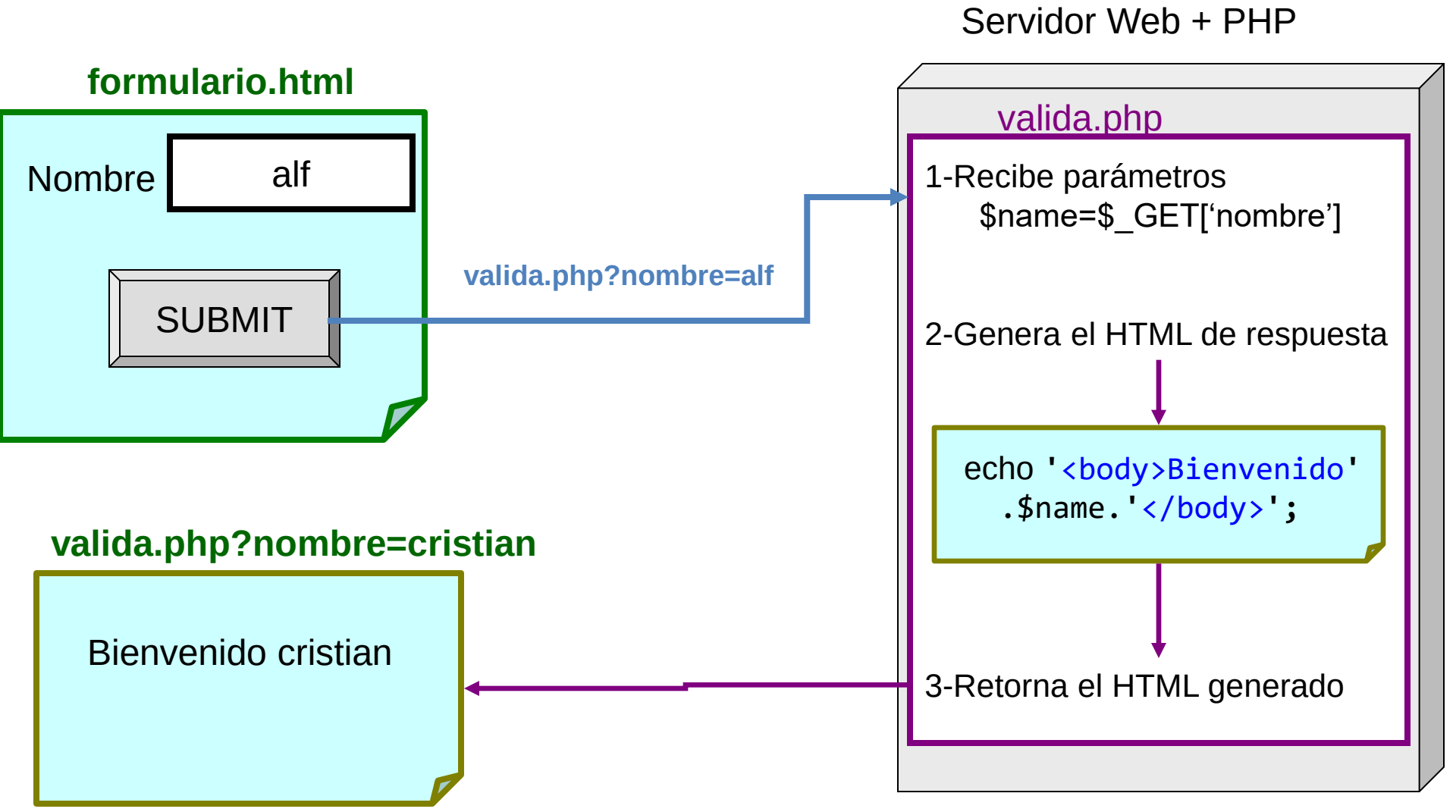


Con Ajax

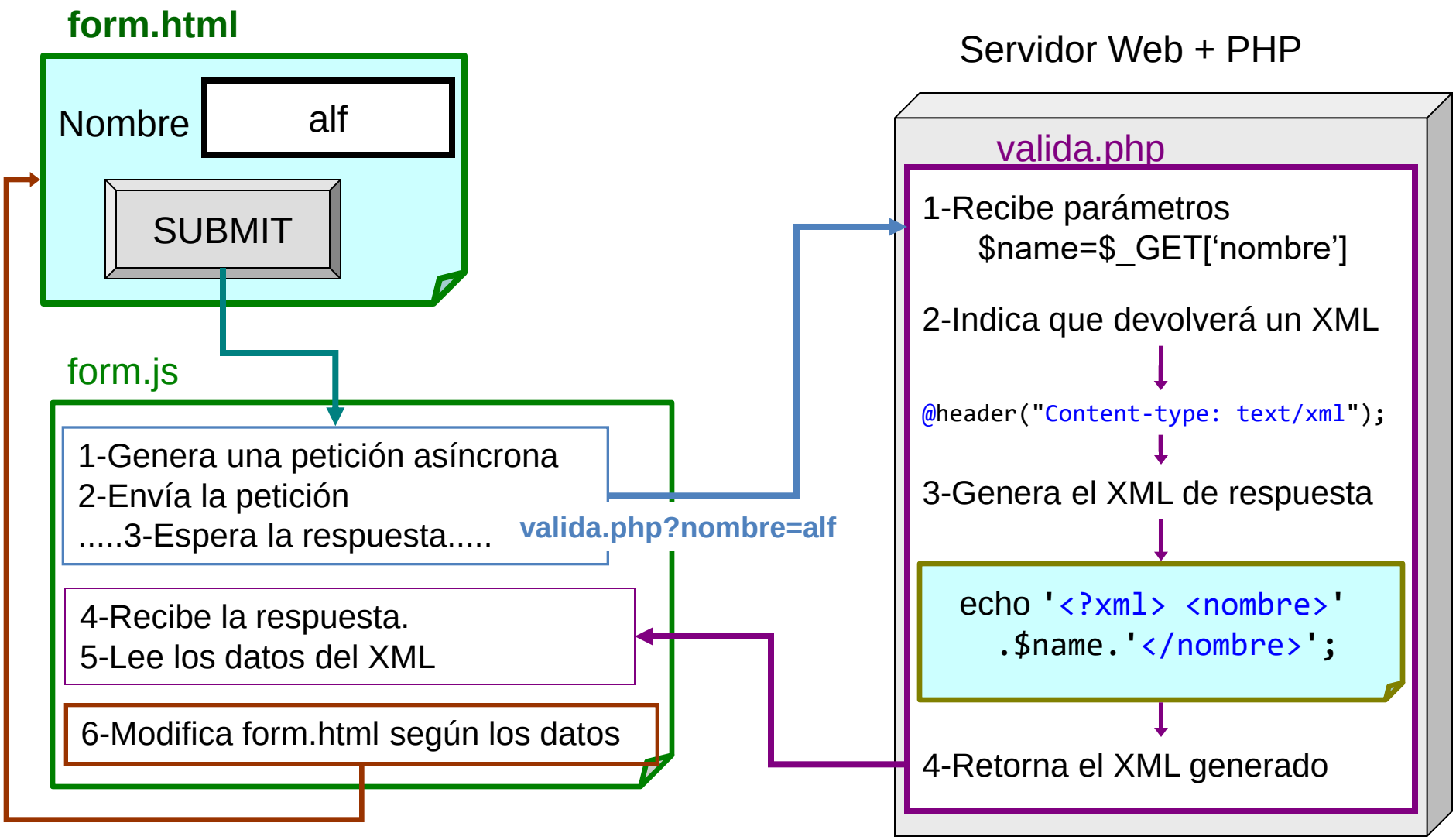
Podemos pedir solo la información que queríamos, no toda la página web.
Necesitaremos a una función que se encargue de insertar los valores pedidos en el HTML



Comunicación sin AJAX (formularios)



Paradigma comunicación con AJAX



Generar una petición GET con AJAX

1-Genera una petición asíncrona
2-Envía la petición
.....3-Espera la respuesta.....

4-Recibe la respuesta.
5-Lee los datos del XML

6-Modifica el HTML según los datos

```
function enviaPeticion(){
   var xmlHttp= new XMLHttpRequest();

   var urlDestino="valida.php?nombre=alf";

  xmlHttp.open("GET", urlDestino, true);

  xmlHttp.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded");
   xmlHttp.onreadystatechange = function(){
       if (xmlHttp.readyState == 4){
          funcionResp(xmlHttp);
       }
    };

    xmlHttp.send(null);
  }
```

funcionResp() la definiremos más a delante

ReadyState

0: sin inicializar
1: abierto
2: cabeceras recibidas
3: cargando
4:completado

Generar la respuesta XML

valida.php

1-Recibe parámetros
$name=$_GET['nombre']

2-Indica que devolverá un XML

```
@header("Content-type: text/xml");
```

3-Genera el XML de respuesta

```
echo '<?xml><nombre>'
  .$name.'</nombre>';
```

4-Retorna el XML generado

valida.php

```
<?php
 $name = $_GET['nombre'];
 //Tipo de archivo a retornar:
 @header("Content-type: text/xml");

 $xml='<?xml version="1.0" encoding="utf-8"?>'."\n";
 $xml.='<nombre><![CDATA['
         .$name.']]></nombre>'."\n";
 // End XML response

 echo ($xml);
?>
```

Recibir la respuesta en formato XML

form.js

1-Genera petición asíncrona
2-Envía la petición
.....3-Espera la respuesta.....

4-Recibe la respuesta.
5-Lee los datos del XML

6-Modifica el HTML

```
function funcionResp(xmlHttp){
  if (xmlHttp.status == 200) {

    var resp = xmlHttp.responseXML;
    var listResp = resp.getElementsByTagName("nombre");
    for (var k = 0; k < listResp.length; k++) {

    var result = listResp[k].childNodes[0].textContent;


      alert(result);
    }
  }
}
```

ReadyState

0:sin inicializar 1: abierto 2: cabeceras recibidas
3: cargando 4:completado

Status

200: OK
404: Page not found

Generar la respuesta JSON

valida.php

1-Recibe parámetros
$name=$_GET['nombre']

3-Genera el JSON de respuesta

4-Retorna el JSON generado

valida.php

```php
<?php
$name = $_GET['nombre'];

$resp='{"nombre":"'.$name.'",
            "nombres":["Maria","Pedro"],
            "info":{"nombres":["nombre1","nombre2"]}
            }';
echo ($resp);
?>
```

- JSON (JavaScript Object Notation) es un lenguaje de estructuracion de datos.
- Formado por parejas : "**nombre**":"**valor**" separadas por comas.
  - "**nombre**":"**valor**" es el valor de "**nombre**"
  - "**nombre**":["**valor1**", "**valor2**"] es una array de 2 elementos("valor1" i "valor2"). *nom[0]* contiene *valor1*, *nom[1]* contiene *valor2*
  - "**nombre**":{"**nom1**":"**valor1**", "**nom2**":"**valor2**"} es una *"array associativa"* de 2 elementos. *nom.nom1* contiene *valor1*, *nom.nom2* contiene *valor2*

Recibir la respuesta en formato JSON

form.js

1-Genera petición asíncrona
2-Envía la petición
.....3-Espera la respuesta.....

4-Recibe la respuesta.
5-Lee los datos del JSON

6-Modifica el HTML

```
function funcionResp(xmlHttp){


    if (xmlHttp.status == 200) {
        var resp = xmlHttp.responseText;

        var respJSON = JSON.parse(resp);

        var txt =respJSON.nombre;
        var nom1=respJSON.nombres[0];
        var nom2=respJSON.nombres[1];
        var color1=respJSON.info.colores[0];
        var color2=respJSON.info.colores[1];

    }
}
```

Estructura del JSON que leemos

```
{"nombre":"'.$name.'",
 "nombres":["Cristian","Pedro"],
 "info":{"colores":["rojo","azul"]}
}
```

Generar una petición POST con AJAX

Se realizan los mismos pasos que en una petición GET excepto:

- Al configurar la petición, se indica que es del tipo **POST**
- Los parámetros no se indican en la **urlDestino** sinó en el **send().**

```
function enviaPeticion(){
   var xmlHttp= getXMLHttpRequest();

   var urlDestino="valida.php";

   xmlHttp.open("POST", urlDestino, true);

   xmlHttp.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded");

   xmlHttp.onreadystatechange = function(){
       if (xmlHttp.readyState == 4){
          funcionResp(xmlHttp);
       }
    };

    xmlHttp.send("nombre=Alf&edad=40");
  }
```

Un poco mas de PHP


```php
<?php
$name = $_GET['nombre'];
//Obtenemos los valores escapando caracteres no deseados
$email = trim(htmlentities(mysql_real_escape_string($_GET["mail"])));
$error=FALSE;    $mensajeError='';
$aleatorio = (rand(1,10)); //aleatorio entre 1 y 10
if(filter_var($email, FILTER_VALIDATE_EMAIL)){
   $error=TRUE;
   $mensajeError.='Email incorrecto:'.$email;
}
if(strlen($name) > 15){//validar el formato del nombre
   $error=TRUE;
   $mensajeError.='el nombre ha de contener menos de 15 caracteres';
 }
@header("Content-type: text/xml");
$xml='<?xml version="1.0" encoding="utf-8"?>'."\n";
$xml.='<xml>'."\n";
if(!$error){
    $xml.='<respuesta><![CDATA['.$name.']]></respuesta>'."\n";
}else{
    $xml.='<error><![CDATA['.$mensajeError.']]></error>'."\n";
}
$xml.='</xml>'."\n";
echo ($xml);

?>
```

ARRAYS PHP

```php
<?php
$nombres= array("Merce","Josefa","Mortadelo","Filemon");//definimos el array
$nombresJSON = json_encode($nombres); //lo pasamos a JSON
//imprimimos el JSON
echo($nombresJSON);  //["Merce","Josefa","Mortadelo","Filemon"]

//definimos array assoc
$pastel=array("nombre"=>"tiramisu", "kcal"=>200,"sabor"=>"rico rico");
$pastelJSON = json_encode($pastel); //lo pasamos a JSON
//lo imprimimos
echo ($pastelJSON);  //{"nombre":"tiramisu","kcal":200,"sabor":"rico rico"}

$jsonReloj='{"marca":"horas","mecanica":"japonesa","precio":"60"}';
$arrayReloj=json_decode($jsonReloj); //generamos el array
var_dump($arrayReloj);  //imprimios el objeto
```


Linkia FP
Formación Profesional
Oficial a Distancia