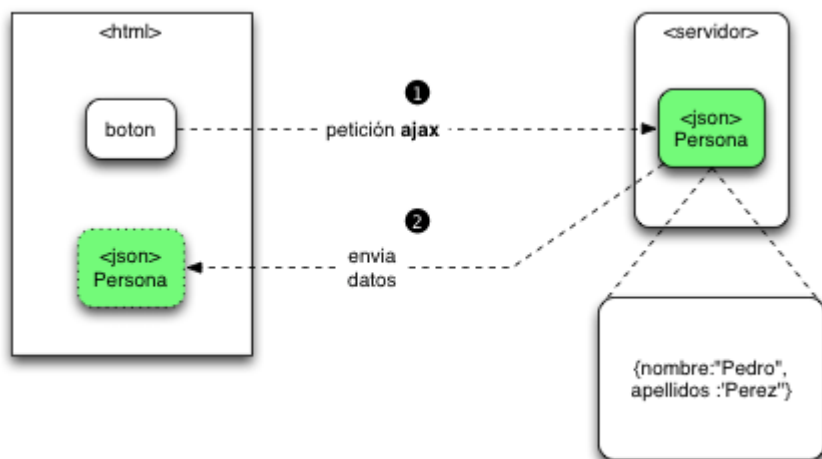
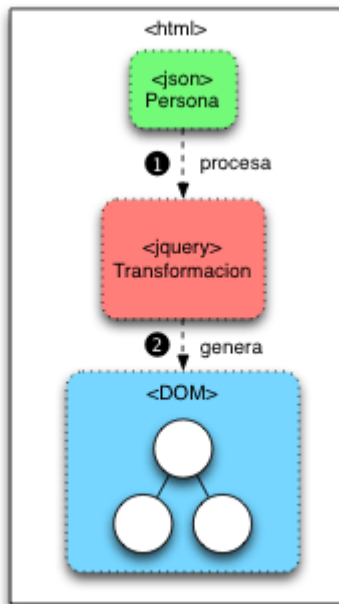


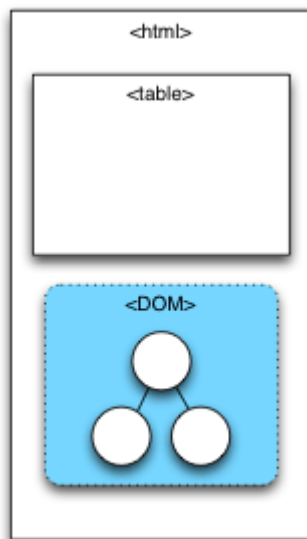
Cuando trabajamos con aplicaciones web en muchos casos nos tenemos que traer estructuras en formato JSON del servidor para después procesarlas y mostrarlas en la pagina como estructura HTML.



Una vez tenemos los datos en formato JSON deberemos realizar un procesamiento para poderlos añadir a la pagina HTML . En muchos casos este procesamiento se realiza con JQuery y se genera una estructura DOM.



Esta estructura será añadida a la página como un nuevo contenido.



Vamos a ver a continuación en código como utilizando JQuery podemos transformar los datos que nos llegan del servidor que en este caso será una lista de personas como la

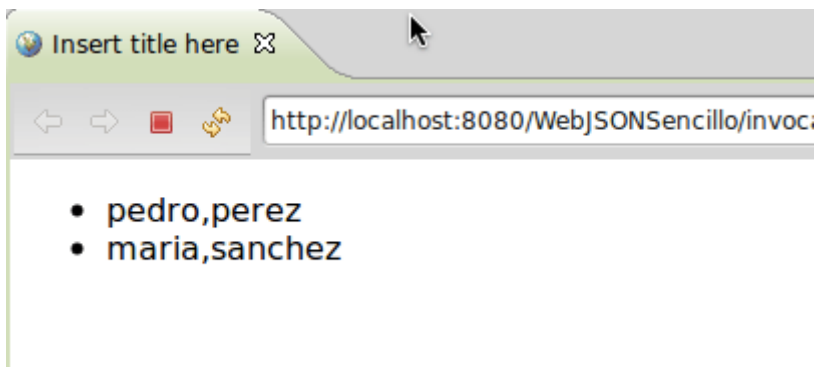
siguiente.

```
{ "personas": [ { "nombre": "pedro", "apellidos": "perez" }, { "nombre": "maria", "apellidos": "sanc  
hez" } ] }
```

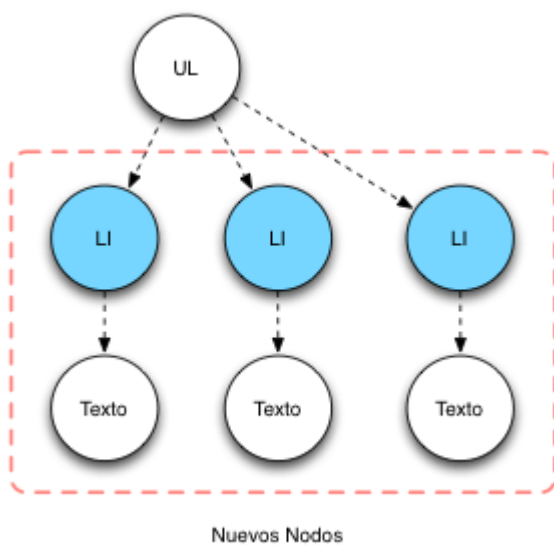
Para recepcionar los datos usamos JQuery y su método \$.getJSON.

```
$(document).ready(function() {  
  
$.getJSON("RecursoJSON",function(datos) {  
$lista=$("");  
$.each( datos.personas, function( clave, valor ) {  
$lista.append("<ul>  
        <li>" + valor.nombre + ", " + valor.apellidos+ "</li>  
</ul>");  
  
});  
    $lista.appendTo("body");  
  
});  
});
```

El código es sencillo de entender simplemente realiza la petición JSON obtiene la lista de personas y con JQuery los convierte en una estructura ul li que añade a la página



Sin embargo tenemos algunos problemas que solventar . En primer lugar estamos construyendo continuamente nodos DOM y añadiendolos a una estructura.



Esta es una de las operaciones mas lentas a la hora de trabajar con DOM y el rendimiento empeorará. Para solventar nuestro problema debemos modificar el código de la forma siguiente.

```
$(document).ready(function() {  
  
$.getJSON("RecursoJSON",function(datos) {
```

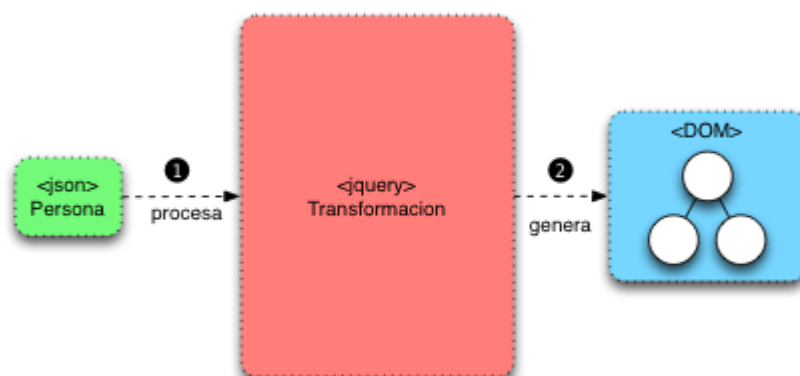
```

$lista=$("");
var cadena="";
$.each( datos.personas, function( clave, valor ) {

cadena+="

```

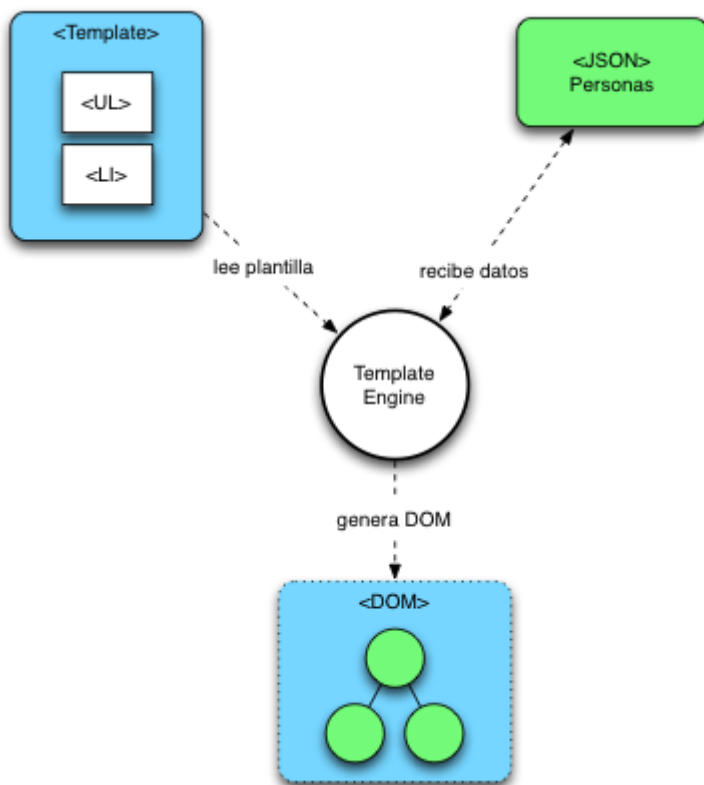
En un ejemplo tan pequeño como el que tenemos no notaremos mucha diferencia pero cuando estemos ante bloques de código complejos la diferencia será clara. Ahora bien todavía nos queda un problema a resolver y es el problema del esfuerzo de programación. Si revisamos nuestro código nos daremos cuenta que lo que más tiempo nos llevará será construir la estructura DOM que queremos añadir a la página a partir de los datos JSON



que nos llegan.

HandleBars Templates

Este problema puede ser solventado a través del uso de un motor de plantillas que nos permite definir una plantilla ,definir los datos en formato JSON y luego procesarlos de forma automática y generar la estructura DOM que necesitamos.



Vamos a verlo construido en código para ello el primer paso será acceder a la página web del motor de plantillas e instalarlo. En este caso he elegido handlebarsjs

<http://handlebarsjs.com/>

Para instalarlo simplemente nos bajaremos el fichero y añadimos el script en nuestra página.

```
<script type="text/javascript" src="handlebars-v1.1.2.js"></script>
```

Una vez hecho esto el siguiente paso es definir un bloque de código con el contenido de la plantilla

```
<script id="plantillaLista" type="text/x-handlebars-template">
<ul>
{{#each personas}}
    <li> {{nombre}} {{apellidos}}</li>

{{/each}}</ul>

</script>
```

La plantilla simplifica de forma clara la forma de trabajar con los datos que nos vuelven en formato JSON. Para ello debemos realizar una serie de modificaciones en nuestro código de Javascript.

```
$(document).ready(function() {

    $.getJSON("RecursoJSON",function(datos) {

        var fuente = $("#plantillaLista").html();
        plantilla = Handlebars.compile(fuente);
        var html = plantilla(datos);
        $("body").append(html);

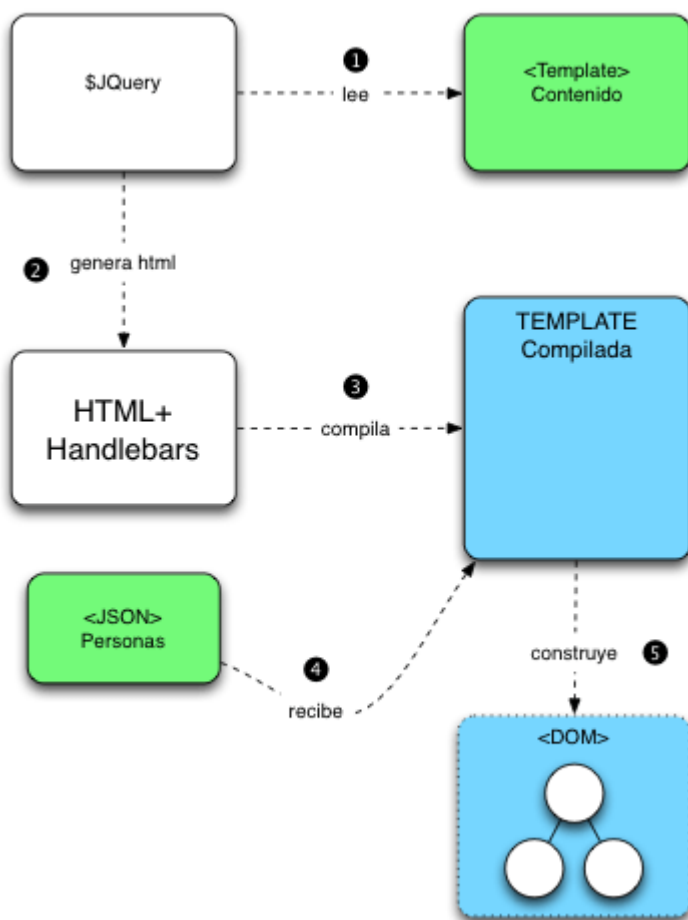
    });
});
```

Los pasos son los siguientes :

1. leo el contenido de la plantilla (fuente)

- Compilo el html fuente y devuelvo una plantilla
- Paso la información de JSON a la plantilla y genero un nuevo HTML
- Añado al arbol DOM el contenido HTML

La siguiente imagen muestra a nivel conceptual los diferentes pasos :



De esta forma las plantillas nos ayudan a reducir el código a construir.