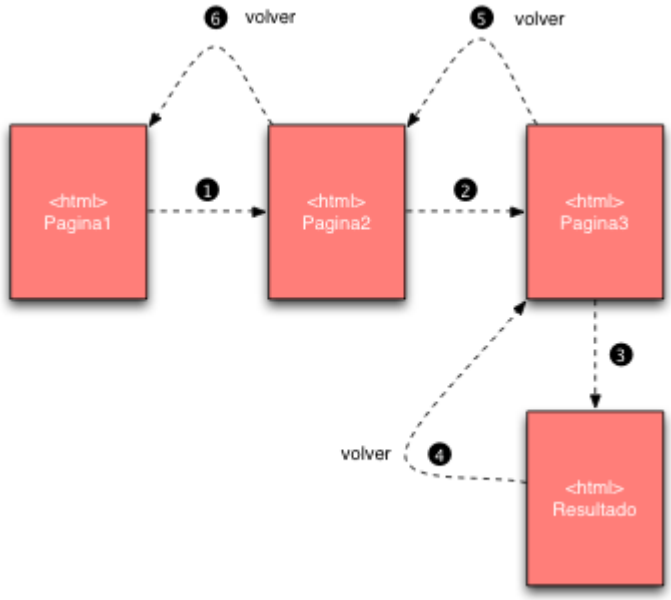
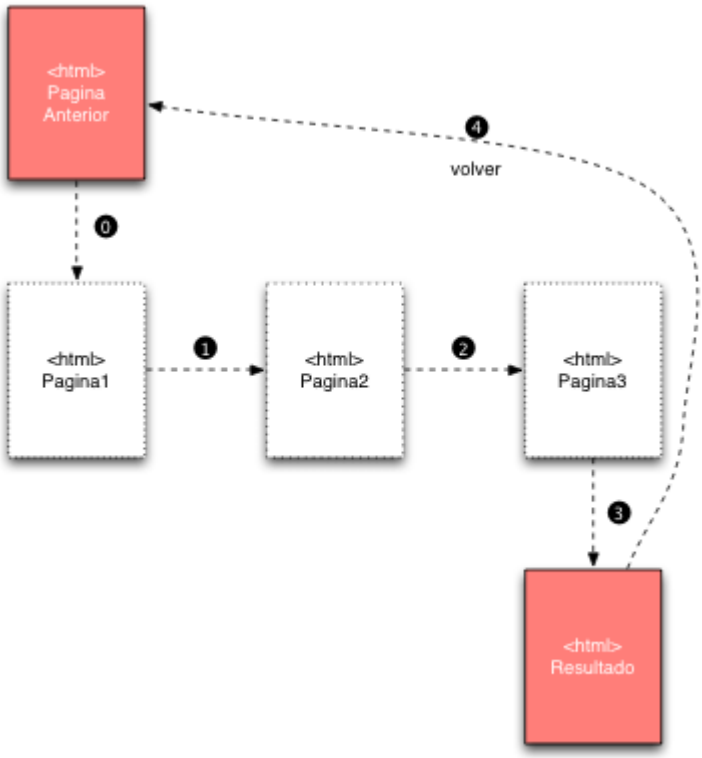
En los artículos anteriores hemos visto como construir una aplicación usando una arquitectura SPA. Sin embargo nos han quedado algunos problemas por resolver uno de los mas típicos es que hacer si el usuario pulsa un botón de volver (back button).



Lamentablemente no conseguiremos de entrada que esto nos funcione correctamente ya que cuando pulsemos la primera vez en el botón de volver nos cargará la página anterior a todos las que están definidas en estructura SPA.



Para solventar este problema deberemos modificar el código construido y hacer uso de HTML 5 y las nuevas capacidades que tiene para manejo del API de historial. Para poder realizar estas operaciones de forma sencilla vamos en un primer lugar a refactorizar un poco el código que tenemos.

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

$("div").hide();
 $("div:first").show();

$("#formulario1").submit(function() {

cambiarPagina("#pagina2","#pagina1");
```

```
return false;

});
 $("#formulario2").submit(function() {
 cambiarPagina("#pagina3","#pagina2");

return false;

});
 $("#formulario3").submit(function() {
 cambiarPagina("#destino","#pagina3");
 return false;

});

});

function cambiarPagina(paginaDestino,paginaOrigen) {

$(paginaOrigen).fadeOut(function() {

$(paginaDestino).fadeIn();
 });

}
```

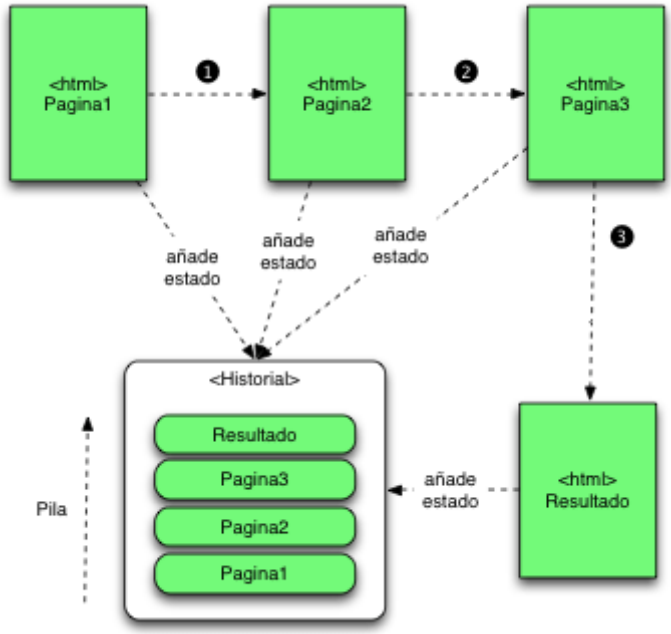
Como vemos simplemente hemos creado una función cambiarPágina para que todo quede mas sencillo.Realizada esta primera operación el siguiente paso será utilizar el API de historial para añadir nuevas estados a cada cambio de página que realizamos dentro de la estructura SPA vamos a verlo.

```
$("div").hide();
 $("div:first").show();
history.pushState({"paginaOrigen":"#pagina1","paginaDestino":"#pagina2
"},"Pagina1","Pagina1.jsp");

 $("#formulario1").submit(function() {

 cambiarPagina("#pagina1","#pagina2");
history.pushState({"paginaOrigen":"#pagina2","paginaDestino":"#pagina3
"},"Pagina2","Pagina2.jsp");

 return false;

});
 $("#formulario2").submit(function() {

 cambiarPagina("#pagina2","#pagina3");
history.pushState({"paginaOrigen":"#pagina3","paginaDestino":"#destino
"},"Pagina3","Pagina3.jsp");

 return false;

});
 $("#formulario3").submit(function() {

history.pushState({"paginaOrigen":"#pagina3","paginaDestino":"#destino
"},"Destino","Destino.jsp");
 cambiarPagina("#pagina3","#destino");

 return false;
```

```
});
```

Acabamos de usar el método del objeto History denominado pushState .Este método nos permite modificar el Historial de nuestro navegador y añadir nuevas entradas

pushState(datos,titulo,URL): Este método tiene tres parámetros .En primer lugar datos que hace referencia a una estructura JSON que podemos pasar como información adicional .En segundo lugar el título de la página y por último la URL asociada. En nuestro caso hemos pasado como datos la pagina origen y destino ya que son los datos que usa nuestra función cambiarPágina.De esta forma hemos conseguido que el historial del navegador quede



modificado.

Para que la aplicación funcione correctamente tendremos ahora que controlar el evento de botón volver (back button) . Encargandonos de leer la estructura JSON que pasamos y cambiar la página que se presenta al usuario.Para ello usaremos el evento “popstate” del navegador apoyándonos en JQuery para manipularlo.

```
$(window).bind("popstate",function(evento) {

if(!evento.originalEvent.state) return;
paginaOrigen=evento.originalEvent.state.paginaOrigen;
paginaDestino=evento.originalEvent.state.paginaDestino;
cambiarPagina(paginaDestino,paginaOrigen);
});
```

El evento popstate en su propiedad “state” almacena la estructura JSON que definimos previamente . Por lo tanto simplemente accedemos a ella e invocamos al método de cambiarPagina.Realizadas estas operaciones nuestra aplicación se comportará de forma correcta en cuanto a la navegación se refiere.