

Categorías

- [Ajax](#) (41)
- [CSS](#) (81)
- [CSS3](#) (29)
- [Destacados](#) (54)
- [Facebook](#) (8)
- [Herramientas](#) (50)
- [HTML](#) (70)
- [HTML5](#) (16)
- [Javascript](#) (58)
- [Joomla](#) (4)
- [jQuery](#) (297)
- [Magento](#) (4)
- [Mootools](#) (7)
- [MySQL](#) (20)
- [Opencart](#) (3)
- [OsCommerce](#) (52)
- [PHP](#) (192)
- [PrestaShop](#) (110)
- [Programation Me](#) (37)
- [Proyectos](#) (6)
- [Redes Sociales](#) (12)
- [Seguridad Web](#) (8)
- [SEO](#) (20)
- [WordPress](#) (51)

APRENDE
WordPress





Crea un tema desde 0

EMPEZAR AHORA

Sígueme en Twitter

[@JoseAguilarBlog](#)



Funciones útiles para utilizar en jQuery



Islas Canarias en Ferry
Disfruta de la playa y la montaña por Canarias, ¡compra tu billete y muévete en Ferry!

COMPRAR



En este artículo pretendemos mostrar una lista de funciones imprescindibles para utilizarlas dentro de tus scripts jQuery.



addClass()

```
$("#contenido").addClass("nombrecalse");
```

Con la función `addClass()` lo que hacemos añadir una clase ya establecida en nuestro CSS al contenedor con `id="contenido"`. En este caso le estamos añadiendo los estilos de la clase `"nombrecalse"`.

after()

```
$("#contenido").after("<p>hola!</p>");
```

El método `after()` añade contenido html delante de lo que haya dentro del contenedor con `id="contenido"`.

ajax()

```
var dataString = 'id='+service;

$.ajax({
  type: "POST",
  url: "archivo.php",
  data: dataString,
  success: function() {
    ...
  }
});
```

Con el método `ajax()` lo que conseguimos es realizar una llamada a un archivo php sin tener que recargar la página ni hacer una petición al servidor.

append()

```
$("#contenido").append("<p>hola!</p>");
```

El método `append()` añade contenido html al contenedor con `id="contenido"`.

appendTo()

```
$("#contenido2").appendTo("#contenido1");
```

El método `appendTo()` copia o mueve el contenedor con `id="contenido2"` al contenedor con `id="contenido1"`.

attr()

Para mejorar al máximo tu experiencia, esta web utiliza cookies. Si utilizas la web significa que estás de acuerdo con que usemos cookies. Hemos publicado una nueva política de cookies, que deberás leer para entender mejor cuáles son las cookies que utilizamos. [Aceptar](#) [Leer más](#)

Novedades

- [DataTables jQuery plugin](#)
- [Speed Index – Otra métrica para medir el rendimiento Web](#)
- [Web Page Test Optimization](#)
- [PrestaShop para tiendas](#)
- [WordPress para Blogs](#)

Entradas más leídas

- [Accordeon me con jQuery](#) (59.523 lecturas)
- [Div que te sigue con el scroll](#) (54.992 lecturas)
- [Popup me con jQuery](#) (54.539 lecturas)
- [Bootstrap jQuery Modal](#) (53.300 lecturas)
- [Paginación de resultados con jQuery, Ajax y PHP](#) (51.468 lecturas)
- [Loading - Efecto cargando con jQuery Ajax](#) (51.359 lecturas)
- [Combos dependientes con jQuery, Ajax y PHP](#) (50.850 lecturas)
- [Abrir contenido sin recargar la página](#) (50.476 lecturas)
- [Validar formulario con jQuery Validate y procesar con Ajax](#) (50.381 lecturas)
- [Paginación de resultados con PHP](#) (46.240 lecturas)

Entradas más comentadas

- [Comprobar disponibilidad de nombre de usuario en vivo](#) (134)
- [Paginación de resultados con jQuery, Ajax y PHP](#) (125)
- [Paginación de resultados con PHP](#) (100)
- [Autocompletar campo con jQuery, Ajax y PHP](#) (91)
- [Envío de comentarios con jQuery y Ajax](#) (88)

```
$("#contenido").attr("data");
```

```
$("#contenido").attr("title");
```

Con la función `attr()` retorna el valor del atributo que le pases como parámetro de un elemento o contenedor.

before()

```
$("#contenido").before("<p>hola!</p>");
```

El método `before()` añade contenido html detrás de lo que haya dentro del contenedor con `id="contenido"`.

blur()

```
$("#input").blur(function(){  
    ...  
});
```

Al salirse de un campo o input de un formulario ejecutaríamos el código de dentro de la función.

change()

```
$("#select#languages").change(function(){  
    ...  
});
```

Al cambiar de opción en un selector de opciones ejecutaríamos el código de dentro de la función.

click()

```
$("#button").click(function(){  
    ...  
});
```

Al hacer click en el evento o contenedor con `id="button"` ejecutaríamos el código de dentro de la función.

css()

```
$("#contenido").css("background-color", "yellow");
```

Con la función `css()` podemos cambiar estilos al contenedor con `id="contenido"`. En el ejemplo estamos cambiando el color de fondo.

dblclick()

```
$("#button").dblclick(function(){  
    ...  
});
```

Al hacer doble click en el evento o contenedor con `id="button"` ejecutaríamos el código de dentro de la función.

delay(time)

```
$("#contenido").slideUp(300).delay(800).fadeIn(400);
```

Esta función permite retrasar la ejecución de las siguientes que estén en la cola. Es útil para añadir retrasos en una serie de animaciones. En el caso del código de ejemplo, primero hacemos `slideUp`, hacemos una espera y seguidamente hacemos `fadeIn`.

each()

```
$("#li").each(function(){  
    $(this).toggleClass("nombre-clase");  
});
```

- [Combos dependientes con jQuery, Ajax y PHP](#) (80)
- [Abrir contenido sin recargar la página](#) (71)
- [Popup me con jQuery](#) (68)
- [Exportar tabla html a Excel con jquery](#) (64)
- [Crear zona de Widgets para añadir plugins](#) (64)

Con la función `each()` lo que hacemos es iterar una serie de elementos. En el ejemplo estamos iterando una lista a la que a cada elemento le estamos cambiando el estilo o css utilizando la función `toggleClass()`.

empty()

```
$("#contenido").empty();
```

Con `empty()` lo que conseguimos es vaciar el contenido de un contenedor antes de volverlo a rellenar.

fadeIn()

```
$("#contenido").fadeIn();
```

El método `fadeIn()` hace que el elemento que lo recibe aparezca en la página a través del cambio de su opacidad, haciendo una transición suavizada que acaba con el valor de `opacity` 1. Este método sólo podremos observarlo si el elemento sobre el que lo invocamos era total o parcialmente transparente, porque si era opaco al hacer un `fadeIn()` no se advertirá ningún cambio de opacidad.

fadeOut()

```
$("#contenido").fadeOut();
```

Este método hace que el elemento que lo recibe desaparezca de la página a través del cambio de su opacidad, haciendo una transición suavizada que acaba con el valor de `opacity` cero.

fadeTo()

```
$("#contenido").fadeTo("slow", 0.33);
```

El método `fadeTo()` es bastante más versátil, como ya se había adelantado. Para hacer un ejemplo interesante con este método tenemos que ver cómo se le pueden pasar distintos valores de opacidad y para ello hemos creado un campo select con distintos valores.

focus()

```
$("#input").focus(function(){  
    ...  
});
```

Al situarte en un campo o input de un formulario ejecutaríamos el código de dentro de la función.

focusout()

```
$("#input").focusout(function(){  
    ...  
});
```

Al salirse de un campo o input de un formulario ejecutaríamos el código de dentro de la función.

hasClass()

```
$("#contenido").hasClass("nombreclase");
```

Con la función `removeClass()` comprobamos si el contenedor con `id="contenido"` tiene la clase "nombreclase". Devuelve `True` si la tiene.

height()

```
$("#contenido").height("200px");
```

Con la función `height()` modificamos el alto del contenedor con `id="contenido"`.

hide()

```
$("#contenido").hide();
```

Con la función `hide()` lo que hacemos es ocultar el contenido del contenedor con `id="contenido"`.

hover()

```
$('#element').hover(function(){  
    ...  
});
```

Con la función `hover()` lo que hacemos es ejecutar un código tras pasar el ratón por encima del contenedor con `id="element"`.

html()

```
$("#contenido").html("<p>hola!</p>");
```

El método `html()` reemplaza el contenido `html` que haya dentro del contenedor con `id="contenido"` por el que le pasas como parámetro.

load()

```
$("#contenido").load("archivo.php");
```

Función muy útil para cargar contenidos sin necesidad de recargar la página. En el caso del ejemplo, lo que haríamos es cargar el contenido `html` del `"archivo.php"` en el contenedor con `id="contenido"`.

keyup()

```
$("#input").keyup(function () {  
    var value = $(this).val();  
    $("#p").text(value);  
}).keyup();
```

Con `keyup()` capturamos los eventos del teclado. En el ejemplo lo que se escriba en el `input` se va escribiendo también en el párrafo.

mouseleave()

```
$("#contenido").mouseleave(function(){  
    ...  
});
```

Al sacar el ratón de encima del contenedor o elemento con `id="contenido"` ejecutaríamos el código de dentro de la función.

mouseout()

```
$("#contenido").mouseout(function(){  
    ...  
});
```

Al sacar el ratón de encima del contenedor o elemento con `id="contenido"` ejecutaríamos el código de dentro de la función.

mouseover()

```
$("#contenido").mouseover(function(){  
    ...  
});
```

nextAll()

```
$("#contenido").nextAll().remove();
```

El método nextAll() selecciona todos los contenedores siguientes al que tiene id="contenido". En el caso del ejemplo, se eliminarán todos los contenedores siguientes a "#contenido".

next()

```
$("#contenido").next().remove();
```

El método next() selecciona el contenedor siguiente al que tiene id="contenido". En el caso del ejemplo, se eliminará el contenedor siguiente a "#contenido".

parent()

```
$("#contenido").parent();
```

El método parent() retorna un objeto que contiene información del contenedor padre del contenedor con id="contenido".

position()

```
$("#contenido").position().top;
```

```
$("#contenido").position().left;
```

La función position() permite obtener la posición de un elemento en relación a su "padre".

ready()

```
$(document).ready(function(){  
    //Aquí tu código  
});
```

Ready es un método propio de jQuery, que revisa si el DOM está listo para usarse. Es más útil que el window.onload, pues este debe esperar a que todos los elementos de la página estén cargados (como scripts e imágenes) para ejecutar. El "ready", en cambio, espera solo a la estructura.

removeClass()

```
$("#contenido").removeClass("nombreclase");
```

Con la función removeClass() lo que hacemos es quitar una clase o estilo CSS al contenedor con id="contenido". En este caso le estamos quitando los estilos de la clase "nombreclase".

slideDown()

```
$("#contenido").slideDown();
```

Con la función slideDown() lo que hacemos es mostrar hacia abajo el contenido del contenedor con id="contenido" que inicialmente o no podría estar oculto.

slideToggle()

```
$("#contenido").slideToggle();
```

Con la función slideToggle() lo que hacemos es mostrar y ocultar el contenido del contenedor con id="contenido" que inicialmente o no podría estar oculto.

slideUp()

```
$("#contenido").slideUp();
```

Con la función `slideUp()` lo que hacemos es mostrar hacia arriba el contenido del contenedor con `id="contenido"` que inicialmente o no podría estar oculto.

show()

```
$("#contenido").show();
```

Con la función `show()` lo que hacemos es mostrar el contenido del contenedor con `id="contenido"` que inicialmente o no podría estar oculto.

size()

```
$("li").size();
```

La función `size()` en este caso retorna el número de elementos que hay en la lista.

submit()

```
$("#formulario").submit(function() {  
    if ($("#code").val() == "correct") {  
        $.ajax({  
            type: "POST",  
            url: "archivo.php",  
            data: dataString,  
            success: function() {  
                $("#message").empty();  
                $("#message").append('<div>Hemos recibido su  
codigo correctamente</div>');  
                $(this).remove();  
            }  
        });  
    }  
    $("#message").empty();  
    $("#message").append('<div>Codigo no valido!</div>');  
    return false;  
});
```

La función `submit()` se suele utilizar para obtener la información de formularios. En el código anterior vemos un ejemplo de lo que se podría hacer. El formulario debe tener `id="formulario"` y método `POST`.

remove()

```
$("#contenido").remove();
```

Con la función `remove()` lo que hacemos es quitar de la vista el contenido del contenedor con `id="contenido"`.

removeAttr()

```
$("#image").removeAttr("width");
```

Con la función `removeAttr()` lo que hacemos es quitar un atributo del elemento con `id="image"`.

text()

```
$("#contenido").text();
```

Con la función `text()` obtiene el contenido textual del elemento o contenedor con `id="contenido"`.

toggle()

```
$("#contenido").toggle();
```

contenedor con id="contenido".

toggleClass()

```
$("#contenido").toggleClass("nombreclase");
```

Con la función toggleClass() podemos añadir y quitar la clase "nombreclase" al contenedor con id="contenido".

val()

```
$("#input").val();

$('select.foo option:selected').val(); // get the value
from a dropdown select
$('select.foo').val(); // get the value
from a dropdown select even easier
$('input:checkbox:checked').val(); // get the value
from a checked checkbox
$('input:radio[name=bar]:checked').val(); // get the value
from a set of radio buttons
```

Con la función val() obtenemos el valor de los inputs, selects, textareas de nuestros formularios.

width()

```
$("#contenido").width("200px");
```

Con la función width() modificamos el ancho del contenedor con id="contenido".

[Ver más funciones](#)



★★★★★ (23 votos, promedio: 4,35 de 5)





[Me gusta](#) A 2 mil personas les gusta esto. [Regístrate](#) para ver qué les gusta a tus amigos.

Comparte en las redes sociales

◀ 1 ▶ 7

Escrito por [Jose Aguilar](#) - Experto programador Prestashop y Wordpress.

30 respuestas a "Funciones útiles para utilizar en jQuery"

-  **avybardice:**
[10/04/2012 a las 16:34](#)
Muy interesante su informacion!!!!
[Responder](#)
-  **Daviddice:**
[25/03/2013 a las 18:01](#)
Gracias por la info!!!, me viene biennn.
[Responder](#)
-  **Daviddice:**
[25/03/2013 a las 18:02](#)
Gracias por la info!!!, me servirá un montónnn
[Responder](#)
-  **coyondice:**
[31/05/2013 a las 18:09](#)

tengo el siguiente problema cuando utilizo el evento change para recargar el form queda en un ciclo infinito; esto estoy haciendo

```
$(function () {  
  $("#Subcategory_Category_CategoryId").change(function () {  
    $("form").submit();  
  });  
});
```

[Responder](#)

◦  *Cristiandice:*

[04/06/2013 a las 15:40](#)

prueba así

```
$(document).ready(function(){  
  $("select#Subcategory_Category_CategoryId").change(  
    function(){  
      ...  
    });  
  });
```

[Responder](#)

5.  *dixon zamoradice:*

[07/06/2013 a las 4:44](#)

Saludos tal vez me puedan ayudar tengo el siguiente código

```
// esperamos que el DOM cargue  
$(document).ready(function() {  
  $("#txtNOMBRE").css("text-transform","uppercase");  
  $("#txtAPELLIDO").css("text-transform","uppercase");  
  
  // definimos las opciones del plugin AJAX FORM  
  var opciones= {  
    beforeSubmit: mostrarLoader, //funcion que se ejecuta  
    antes de enviar el form  
    success: mostrarRespuesta, //funcion que se ejecuta una vez  
    enviado el formulario  
  
  };  
  
  //asignamos el plugin ajaxForm al formulario y le pasamos  
  las opciones  
  $('#hongkiat-form').ajaxForm(opciones);  
  
  //lugar donde defino las funciones que utilizo dentro de  
  "opciones"  
  function mostrarLoader(){  
    $("#loader_gif").fadeIn("slow");  
  };  
  function mostrarRespuesta (responseText){  
    alert("Mensaje enviado: "+responseText); //responseText es  
    lo que devuelve la página contacto.php. Si en contacto.php  
    hacemos echo "Hola" , la variable responseText = "Hola" .  
    Aca hago un alert con el valor de response text  
    $("#loader_gif").fadeOut("slow"); // Hago desaparecer el  
    loader de ajax  
    //$("#ajax_loader").append("Mensaje: "+responseText); //  
    Aca utilizo la función append de JQuery para añadir el  
    responseText dentro del div "ajax_loader"  
  };  
});
```

para guardar en una base de datos y lo hace pero ahora

~~como hago para buscar esos datos y mostrarlos en el formulario gracias~~

[Responder](#)

6.  Enzodice:

[04/08/2013 a las 12:56](#)

Te pasaste, venia buscando esto hace tiempo!

[Responder](#)

7.  toqu3dice:

[07/10/2013 a las 18:34](#)

Like 😊

[Responder](#)

8.  carlos julio muñoz lopezdice:

[06/10/2014 a las 13:11](#)

Cordial saludo, agradezco el que quieras compartir tus conocimientos.

Te quiero hacer la siguiente consulta: Tengo una aplicación web hecha en lenguaje "go", necesito saber como hago para hacer una llamada a una función del lado del servidor, después de un evento en un campo de un formulario.

Después de actualizar o perder el foco, por ejemplo, necesito que se llame a una función en servidor para que se haga alguna verificación u operación y poder tomar este resultado para hacer algo en la pagina.

Si tienes algún ejemplo o me indicas donde encontrarlo. Mil gracias!!

[Responder](#)

◦  Jose Aguilardice:

[06/10/2014 a las 19:45](#)

Posiblemente te pueda servir este ejemplo:

<http://www.jose-aguilar.com/blog/comprobar-disponibilidad-de-nombre-de-usuario-en-vivo/>

Saludos

[Responder](#)

▪  carlos julio muñoz lopezdice:

[08/10/2014 a las 13:08](#)

Mil gracias!!! , me ha servido tu ejemplo. Claro conciso y efectivo!!!!


[Responder](#)

9.  Alejandrodice:

[16/10/2014 a las 2:43](#)

Hola, te hago una preguntar tengo un combobox estoy tratando de hacer que cuando se haga un hover sobre una opcion a un costado muestre una imgen, he intentado pero no me ha salido, me podes ayudar. Gracias

[Responder](#)

10.  Jose Garciadice:

[25/11/2014 a las 6:34](#)

Hola

A ver si me podeis ayudar, es posible llamar a varias funciones desde un solo evento? en este caso quiero hacer una llamada a tres funciones con el evento onkeyup. Obviamente el codigo que adlunto no me funciona.

Saludos y Gracias

[Responder](#)

Hola, yo diría que si que dentro del onkeyup puedes llamar a las funciones que quieras...

[Responder](#)

11. *Efectos, Selectores, Eventos y Funciones de JQuery | Mis publicaciones de Diseño Gráfico* dice:
[19/01/2015 a las 6:10](#)
[...] <http://www.jose-aguilar.com/blog/funciones-utiles-para-utilizar-en-jquery/>, recuperado el 16 de enero de 2015.
[...]


[Responder](#)

12. *Trabajo Grupal | paulimas* dice:
[23/01/2015 a las 3:19](#)
[...] <http://www.jose-aguilar.com/blog/funciones-utiles-para-utilizar-en-jquery/>, recuperado el 16 de enero de 2015.
[...]


[Responder](#)

13. *Efectos, Selectores, Eventos y Funciones de JQuery | rxbxrt's Bloggers* dice:
[26/01/2015 a las 19:20](#)
[...] <http://www.jose-aguilar.com/blog/funciones-utiles-para-utilizar-en-jquery/>, recuperado el 16 de enero de 2015.
[...]


[Responder](#)

14.  *kristian* dice:
[17/08/2015 a las 13:46](#)
muy buena la informacion de aqui, muchas gracias

[Responder](#)


15.  *Sergio Daniel Calvo Hidalgo* dice:
[16/10/2015 a las 12:55](#)
Sobre la funcion .appenTo(), existe otra función parecida llamada .prependTo(), que hace justo lo contrario, en vez de colocar las cosas al final, las coloca justo al principio. Buena entrada 😊

[Responder](#)

16.  *sild* dice:
[19/01/2016 a las 10:58](#)
Hola. Existe alguna manera para que al hacer un slideToggle, el contenido de la pagina no se desplace?. Ahora mismo tengo un botón de info que al hacer hover se despliega información adicional, pero todo el contenido que tengo por debajo se desplaza para dejar hueco al texto adicional. Se puede hacer que no se desplace?

muchas gracias

[Responder](#)

-  *Jose Aguilar* dice:
[19/01/2016 a las 12:27](#)

Hola,

Quizás te pueda interesar este post:

<http://www.jose-aguilar.com/blog/tooltips-click/>

Saludos

[Responder](#)

¡Hola! Me llamo Laura y soy nueva en jQuery. Estoy haciendo unos ejercicios y no consigo hacer este : con jQuery, mostrar en pantalla un conjunto de números y al pulsar el botón "ver resultado" seleccionar en la pantalla únicamente los números impares.

Alguien me puede ayudar? Gracias!!

[Responder](#)

◦  Vicentedice:

[30/08/2017 a las 10:20](#)

Al final pudiste hacer este ejercicio? Creo que estoy haciendo el mismo ejercicio que tú, je je. Si puedes me dices. Gracias.

[Responder](#)

18.  Sebastiandice:

[03/05/2016 a las 13:33](#)

excelente post! gracias por tu tiempo!

[Responder](#)

19.  Javierdice:

[25/09/2016 a las 11:24](#)

Hola !

Es muy practico y util este blog. Gracias por compartir.

En Prestashop quiero que se ejecute en la pantalla de login

\$ (".account-creation").append("jgprueba") ;

Para que añada a la clase account-creation el parrafo que indico.

No se que estare haciendo mal porque no funciona.

Si podeis ayudarme.

Gracias

[Responder](#)

◦  Javierdice:

[25/09/2016 a las 11:25](#)

Es en la segunda pantalla donde aparece el boton registrar.

[Responder](#)

◦  Jose Aguilar dice:

[30/09/2016 a las 12:00](#)

Hola,

No se si ya habrás podido solucionar este problema pero tienes que buscar el sitio correcto para ejecutar tu código:

\$ (".account-creation").append("jgprueba") ;

Saludos

[Responder](#)

20.  Pablo Mendez Lizarbedice:

[22/10/2017 a las 15:51](#)

Gran ayuda, lo más utilizado.

[Responder](#)

21.  Mariodice:

[14/12/2017 a las 12:36](#)

Hola es la primera vez que entro esta web y me he decidido a comentar. Me gusta este blog. ¿thèmeutiliza?
me gustaria poder utlizarlo para mi sitio pero no lo encuentro.
¿Es algún CMS como Budypress?

Si no molesta, no encuentro ningún marcador social como Digg creo que deberiais tener alguno.
Yo tengo Pinterest dado que es facil de usar

[Responder](#)

◦  [Jose Aguilar](#) dice:

[16/12/2017 a las 8:49](#)

Hola buenos días,

El tema que estamos utilizando para este blog es personalizado y realizado a medida. Sentimos decirle que no está disponible para la descarga.

Saludos

[Responder](#)

Deja un comentario

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con *

Comentario

Nombre *

Correo electrónico *

Web

☐

Guardar mi nombre, correo electrónico y sitio web en este navegador para la próxima vez que haga un comentario.

Este sitio usa Akismet para reducir el spam. [Aprende cómo se procesan los datos de tus comentarios.](#)

0 artículos que te podrían interesar

Ver más sobre [funciones](#)



Jose Aguilar

Ingeniero técnico en
Informática de Gestión
analista programador de

**Categorías
Principales**

- [jQuery](#)
- [PHP](#)
- [Ajax](#)
- [Programación Me](#)

En este blog se habla de...

[accordiones \(12\)](#) [archivos \(31\)](#) [arrays \(10\)](#)
[bootstrap \(9\)](#) [botones \(11\)](#) [buscadores \(7\)](#) [carousel \(12\)](#)
[checkboxes \(15\)](#) [classes \(5\)](#) [contenedores](#)
[\(71\)](#) [countdown \(14\)](#) [Crear un tema WordPress](#)

Para mejorar al máximo tu experiencia, esta web utiliza cookies. Si utilizas la web significa que estás de acuerdo con que usemos cookies. Hemos publicado una nueva política de cookies, que deberás leer para entender mejor cuáles son las cookies que utilizamos. [Aceptar](#) [Leer más](#)

Prestashop y experto
programador de blogs y
páginas Webs con
Wordpress.

[Leer más](#)

- [PrestaShop](#)
- [WordPress](#)

[desde cero \(12\)](#) [datepickerers \(11\)](#) [dialogs \(13\)](#) [editores html](#)

[\(6\)](#) [emails \(6\)](#) [eventos \(40\)](#) [fechas \(9\)](#)

[formularios \(93\)](#) [framework \(6\)](#) [galerias \(17\)](#)

[imágenes \(52\)](#) [inputs \(43\)](#) [lightbox \(16\)](#)

[links \(7\)](#) [marketplace \(9\)](#) [menús \(35\)](#) [módulos](#)

[prestashop \(51\)](#) [paginación \(5\)](#) [password \(5\)](#) [popups](#)

[\(20\)](#) [programas \(11\)](#) [radiobutton \(9\)](#) [ratings \(6\)](#) [registros](#)

[\(15\)](#) [scroll \(12\)](#) [seguridad \(5\)](#) [selects \(27\)](#) [sliders](#)

[\(40\)](#) [tabs \(9\)](#) [textarea \(12\)](#) [textos \(9\)](#) [tooltips \(12\)](#)

[validaciones \(20\)](#) [zoom \(18\)](#)

Copyright © 2008-2018 - [Jose Aguilar](#) | [f](#) [t](#) [in](#) [g+](#) [You](#) [RSS](#)

