



JavaScript



**ANEXO 8.-**  
jQueryUI



**DISTRIBUIDO POR:**

**CENTRO DE INFORMÁTICA PROFESIONAL S.L.**

C/ URGELL, 100  
08011 BARCELONA  
TFNO: 93 426 50 87

C/ RAFAELA YBARRA, 10  
48014 BILBAO  
TFNO: 94 448 31 33

[www.cipsa.net](http://www.cipsa.net)

**RESERVADOS TODOS LOS DERECHOS. QUEDA PROHIBIDO TODO TIPO DE  
REPRODUCCIÓN TOTAL O PARCIAL DE ESTE MANUAL, SIN PREVIO  
CONSENTIMIENTO POR EL ESCRITOR DEL EDITOR**

## JQuery UI

JQuery UI es una librería de Javascript creada a partir de JQuery que provee de efectos, temas, eventos y componentes (*widgets*) para mejorar el aspecto y funcionalidad de páginas y aplicaciones web.

Al igual que en el caso de JQuery, JQuery UI puede instalarse de dos modos:

### Descarga de librería personalizada

Es posible descargar una versión personalizada de la librería para integrarla como un recurso más de nuestra aplicación desde la web oficial:

<http://jqueryui.com/download/>

La página consta de un generador de descargar que permite seleccionarse la versión de JQueryUI que queremos, (la cual depende de la versión de JQuery que usamos), junto con los eventos, temas, efectos y widgets que nos interesan para descargar una versión personalizada de la librería con el tamaño justo.

### Integración con CDN (*Content Delivery Network*)

En este caso podemos hacer referencia a la última versión de la librería en versiones no comprimida y comprimida. Los enlaces se encuentran en la web oficial de JQuery:

<https://code.jquery.com/ui/>

En la página pueden encontrarse los enlaces a la última versión de la librería comprimida (*minified*) y sin comprimir (*uncompressed*). Al pulsar sobre los enlaces se muestra el código que debemos añadir en nuestras páginas para integrar la librería.

### Temas (*Themes*)

La librería JQueryUI requiere de un tema para funcionar. Los temas son hojas de estilos CSS que definen las formas, colores y diseños de los *widgets*.

Al igual que con la propia librería, existen temas predefinidos que podemos descargar de la web oficial de JQuery UI, o crear nuestros propios estilos personalizados mediante la herramienta de generación (*ThemeRoller*):

<http://jqueryui.com/themeroller/>

También podemos integrar los temas predefinidos desde la web oficial de JQuery:

<https://code.jquery.com/ui/>

### Resumiendo

Para poder utilizar JQueryUI en las páginas web de nuestro sitio o aplicación web es preciso añadir las siguientes referencias:

- Librería javascript de *JQuery* (*jquery-x.x.x.js*)
- Librería javascript de *JQueryUI* (*jquery-ui-x.x.x.js*)
- Hoja de estilos del tema para *JQuery* (*theme.css*)

**Ejemplo:** Código de página con *JQuery*, *JQueryUI*, y tema básico integrados.

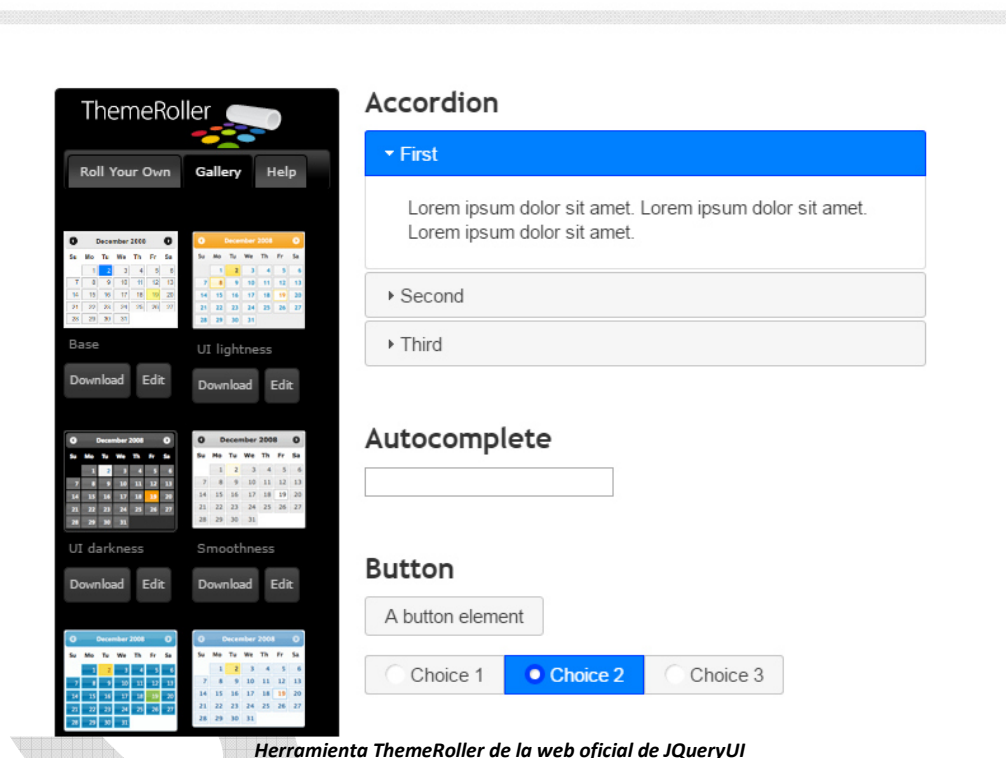
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css"
        rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
          integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtiJj06wt/s="
          crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
          integrity="sha256-eGE6blurk5sHj+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
          crossorigin="anonymous"></script>
  <script>
</head>
<body>
</body>
</html>
```

## Componentes (*Widgets*)

Los componentes son elementos definidos por JQueryUI que permiten organizar el contenido o solicitar datos al usuario de manera más flexible y cómoda sin necesidad de tener que implementarlos por nosotros mismos. La apariencia de los componentes depende del tema, por lo que debemos emplear el más cercano al estilo de nuestra página o definir uno personalizado.

(\*) En la página web oficial de JQueryUI, en el apartado Themes, podemos emplear la herramienta ThemeRoller para elegir un tema predeterminado de la galería (*Gallery*) o crear uno personalizado (*Roll your Own*) viendo el aspecto de los componentes.

## ThemeRoller



Herramienta ThemeRoller de la web oficial de JQueryUI

La mayoría de los widgets se basan en elementos HTML. Para convertirlos en widgets deben seleccionarse desde javascript e invocar a una función determinada que los convierte en widgets.

**Ejemplo:** El siguiente código muestra la declaración de un *spinner* a partir de un control `<input>` de HTML:

```
<script>
$(function () {
    var widget_spinner = $("#spinner").spinner();
});
</script>
<p>
<label for="spinner">Select a value:</label>
<input id="spinner" name="value">
</p>
```

Todos los widgets pueden personalizarse permitiendo modificar su aspecto y comportamiento mediante un objeto cuyos atributos especifican las diferentes características del widget y que se pasa como argumento a la función del widget:

```
<script>
  $(function () {
    var widget_spinner = $("#spinner").spinner(
      {
        min: 5,           // Parámetros de configuración para el Widget
        max: 2500,
        step: 25,
        start: 1000
      });
  });
</script>

<p>
  <label for="spinner">Select a value:</label>
  <input id="spinner" name="value">
</p>
```

Los atributos del objeto de configuración para de cada widget son variados y dependen de las características de cada uno. Estos atributos pueden consultarse en la documentacion de la API en la web oficial de JQueryUI:

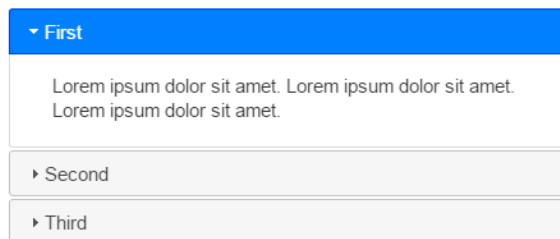
<http://api.jqueryui.com/>

A continuación se muestran algunos widgets interesantes y las opciones más comunes de configuración para su uso.

**Acordeón ( Accordion )**

<http://jqueryui.com/accordion/>

Este componente representa una serie de capas que pueden plegarse sobre si mismas para mostrar diferentes contenidos en un espacio limitado.



*Widget acordeón de tres capas con estilo básico*

Cada capa va asociada a una etiqueta en la parte principal que al ser pulsada provoca que se despliegue o repliegue la capa mostrando u ocultando su contenido. La implementación se realiza definiendo una capa que contendrá las capas con cada uno de los contenidos. Esa capa debe seleccionarse desde código e invocar la función. **accordion()**:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title></title>

  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css"
        rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
          integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtiJj06wt/s="
          crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
          integrity="sha256-eGE6blurk5sHj+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
          crossorigin="anonymous"></script>

  <script>
    $( function() { $( "#accordion" ).accordion(); });
  </script>
</head>
<body>
  <div id="accordion">
    </div>
</body>
</html>
```

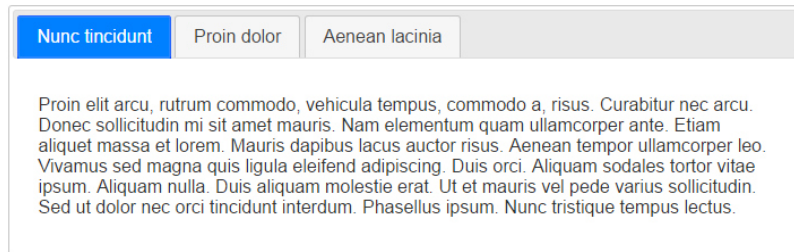
En la capa “*accordion*” podemos añadir tantas capas de contenidos como deseemos con un párrafo inicial que determina el título de la capa y el contenido siguiente.

```
<h3>Primera sección</h3>
<div>
  <p>
    .. contenido ..
  </p>
</div>
```

## Pestañas ( Tabs )

<http://jqueryui.com/tabs/>

Las pestañas son un conjunto de botones que permiten cambiar entre diferentes contenidos en una misma área de la página.



Widget pestañas con tema básico.

Este widget se crea a partir de una capa. Esta debe seleccionarse desde javascript e invocar a la función ***tabs()***:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cUeBR6CsyA4/9szpPfrX3s49M9vUU5BgtlJ06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurk5sHj+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>
  <script>
    $( function() { $("#tabs").tabs(); });
  </script>
</head>
<body>
  <div id="tabs">
    <div>
  </div>
</body>
</html>
```

La capa que contiene el widget debe tener una lista de enlaces con el nombre de cada pestaña y una referencia al atributo “id” de la capa con el contenido correspondiente:

```
<!-- lista de pestañas de contenidos -->
<ul>
  <li><a href="#tabs-1">Contenido 1</a></li>
  <li><a href="#tabs-2">Contenido 2</a></li>
  <li><a href="#tabs-3">Contenido 3</a></li>
</ul>
<!-- capas con contenidos -->
<div id="tabs-1">
  .. contenido ..
</div>
<div id="tabs-2">
  .. contenido ..
</div>
<div id="tabs-3">
  .. contenido ..
</div>
```



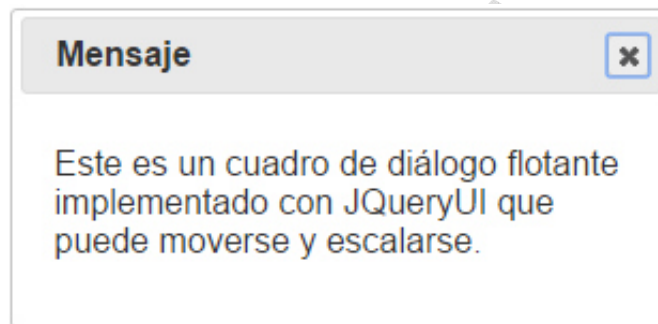
Cuadros de diálogo (*Dialog*)

<http://jqueryui.com/dialog/>

Este widget muestra un cuadro flotante sobre la página con un contenido. Puede emplearse para mostrar al usuario un mensaje, solicitarle una confirmación, o solicitar unos datos presentando un formulario al completo.

(\*) La funcionalidad de este widget es semejante a la de las funciones *alert()*, *confirm()* y *prompt()* de Javascript, pero permite personalizar el contenido del cuadro de diálogo.

(\*) Este widget no abre una nueva ventana del navegador; muestra el contenido como un cuadro flotante dentro de la propia página.



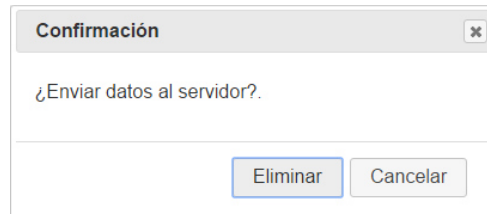
Widget cuadro de diálogo con tema básico

La implementación de este widget se realiza a partir de una capa a la que selecciona desde javascript e invoca la función ***dialog()***, para mostrar su contenido como un cuadro de diálogo. El título del cuadro se toma del valor del atributo *"title"* de la capa.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title></title>
  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtiJj06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurk5sHj+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>
  <script>
    $( function() { $( "#dialog" ).dialog(); } );
  </script>
</head>
<body>
  <div id="dialog" title="Mensaje">
    .. contenido del cuadro de diálogo ..
  </div>
</body>
</html>
```

La función **dialog()** puede recibir un objeto con diferentes parámetros que determinen ciertos aspectos del cuadro de diálogo mostrado.

**Ejemplo:** El siguiente código muestra un cuadro de diálogo acompañado de dos botones que el usuario puede pulsar para indicar una respuesta:



Widget de cuadro de diálogo con botones con estilo básico

```
$( function() {  
    $("#dialog").dialog(  
    {  
        resizable: false,  
        height: "auto",  
        width: 400,  
        modal: true,  
        buttons:  
        {  
            "Eliminar": function () {  
                $(this).dialog("close");  
            },  
            "Cancelar": function () {  
                $(this).dialog("close");  
            }  
        }  
    });  
});
```

El objeto indicado como argumento para la función dialog puede constar de los siguientes atributos:

- **resizable** → Indica si el cuadro de diálogo puede reescalarse.
- **width, height** → Indica la altura y anchura del cuadro de diálogo.
- **modal** → Indica si el cuadro de diálogo es modal o no.
- **buttons** → indica qué botones deben mostrarse y la acciones a realizar al ser pulsados.

(\*) Un cuadro de diálogo modal no permite la interacción con el resto de elementos de la página hasta que es cerrado exigiendo una respuesta al usuario. En ese caso la página aparece en tono gris indicando que el usuario sólo puede interactuar con el cuadro de diálogo mostrado.

El atributo "**buttons**" define a su vez un objeto con los botones que deseamos mostrar como parte del cuadro de diálogo. Por cada botón puede indicarse como mínimo el texto mostrado y la acción a realizar al ser pulsado.

**Ejemplo:** Este código define un botón con el mensaje “Eliminar” que al pulsarse cierra el cuadro de diálogo llamando a la función **dialog()** con el argumento “close”.

```
"Eliminar": function () {
    // PROCESADO DE DATOS
    $(this).dialog("close");
},
```

También es posible mostrar formularios al completo en el cuadro de diálogo para solicitar y recoger datos sin cambiar de página.

**Ejemplo:** Supóngase una página que muestra una lista de personas a las que el usuario puede añadir nuevos elementos pulsando el botón “Nuevo”:

NOMBRE	E-MAIL
roger	none@none.com

Nuevo

```
<!-- Lista de usuarios registrados -->
<div id="contenedor_usuarios">
  <table id="usuarios">
    <thead>
      <tr>
        <th>NOMBRE</th>
        <th>E-MAIL</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>roger</td>
        <td>none@none.com</td>
      </tr>
    </tbody>
  </table>
</div>
<!-- Boton para visualizar cuadro de diálogo -->
<input type="button" id="nuevo_usuario" value="Nuevo" />
```

Al pulsarse el botón “**Nuevo**” (*id*="nuevo\_usuario") debe mostrarse un cuadro de diálogo modal solicitando los datos de la nueva persona a registrar:

Nuevo usuario

Nombre

Correo

Registrar

Widget de cuadro de diálogo con formulario

El cuadro de diálogo se crea a partir de una capa con los elementos del formulario:

```
<!-- Capa formulario del cuadro de dialogo -->
<div id="dialogo" title="Nuevo usuario">
  <form>
    <label for="nombre">Nombre</label><br />
    <input type="text" id="nombre"/><br />
    <label for="correo">Correo</label><br />
    <input type="email" id="correo"/><br />
  </form>
</div>
```

Para ello se define el formulario en una capa con *id="dialogo"* que se establece como cuadro de diálogo seleccionando la capa y llamando al método *dialog()* desde Javascript.

Para que el cuadro de diálogo no se muestre de manera predeterminada al abrirse la página debe indicarse al atributo *"autoOpen"* con el valor *"false"* en la función ***dialog()***. Debe además guardarse en una variable la referencia al cuadro de diálogo para poder mostrarlo y ocultarlo después invocando a la función ***dialog()*** indicando los valores *"open"* y *"close"* como argumento respectivamente.

El cuadro de diálogo incluye la declaración de un botón *"Registrar"* que al ser pulsado procesa los datos y oculta el cuadro de diálogo. Al pulsarse este botón debe añadirse los datos de la persona indicada a la lista invocando a la función ***agregar()***:

```
cuadro_dialogo = $("#dialogo").dialog({
  autoOpen: false,
  height: "auto",
  width: "auto",
  modal: true,
  buttons: {
    "Registrar": function () {
      agregar();
      $(this).dialog("close"); // Oculta cuadro de dialogo
    }
  }
});
```

**Código completo:**

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title></title>

  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtiJj06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurk5sHj+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>

  <script>
    // Funcion de inicializacion de Javascript.
    $(function () {
      // Referencia al cuadro de dialogo
      var cuadro_dialogo;
```

```

// Agrega nuevo usuario a la lista
function agregar() {
    $("#usuarios tbody").append("<tr>" +
        "<td>" + $("#nombre").val() + "</td>" +
        "<td>" + $("#correo").val() + "</td>" +
        "</tr>");
}

// Declaracion del cuadro de dialogo.
cuadro_dialogo = $("#dialogo").dialog({
    autoOpen: false, // No se muestra al principio.
    height: "auto",
    width: "auto",
    modal: true,
    buttons: {
        "Registrar": function () {
            agregar();
            $(this).dialog("close"); // Cierra del cuadro de diálogo
        }
    }
});

// Pulsacion de boton "NUEVO"
$("#nuevo_usuario").on("click", function () {
    cuadro_dialogo.dialog("open"); // Muestra el cuadro de diálogo
});
});
</script>
</head>
<body>
<!-- Capa formulario del cuadro de dialogo -->
<div id="dialogo" title="Nuevo usuario">
    <form>
        <label for="nombre">Nombre</label><br />
        <input type="text" id="nombre"/><br />
        <label for="correo">Correo</label><br />
        <input type="email" id="correo"/><br />
    </form>
</div>

<!-- Lista de usuarios registrados -->
<div id="contenedor_usuarios">
    <table id="usuarios">
        <thead>
            <tr>
                <th>NOMBRE</th>
                <th>E-MAIL</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>roger</td>
                <td>none@none.com</td>
            </tr>
        </tbody>
    </table>
</div>

<!-- Boton para visualizar cuadro de diálogo -->
<input type="button" id="nuevo_usuario" value="Nuevo" />
</body>
</html>

```

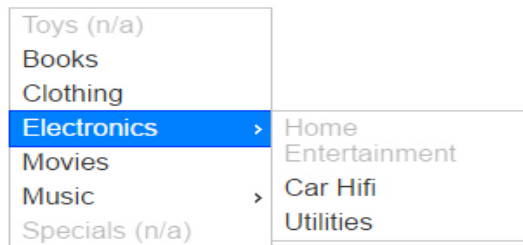
Para un ejemplo completo incluyendo validación de los campos del formulario del cuadro de diálogo puede examinarse el ejemplo de la web oficial de JQueryUI:

<http://jqueryui.com/dialog/#modal-form>

## Menu

<http://jqueryui.com/menu/>

Este widget muestra un menu de opciones en el que pueden seleccionarse opciones y desplegar submenús.



Widget Menu con estilo básico.

El componente se crea a partir de una lista `<ul>` que debe seleccionarse mediante javascript y llamar a la función ***menu()***.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title></title>

  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vU5BgtiJj06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurk5sHj+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>

  <script>
    $( function() {
      $( "#menu" ).menu();
    } );
  </script>

  <style>
    /* Ancho del menú desplegable */
    .ui-menu {
      width: 150px;
    }
  </style>
</head>
<body>
  <ul id="menu">
  </ul>
</body>
</html>
```

En la lista deben indicarse los diferentes elementos que conforman las opciones del menú:

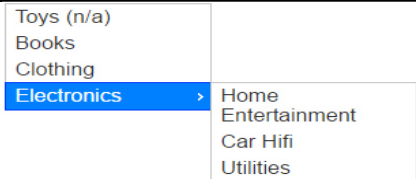
```
<ul id="menu">
  <li><div>Toys (n/a)</div></li>
  <li><div>Books</div></li>
  <li><div>Clothing</div></li>
</ul>
```

```
Toys (n/a)
Books
Clothing
```

## CURSO DE PROGRAMACION JAVASCRIPT

Para indicar submenús sólo es necesario declarar un elemento `<ul>` anidado en un elemento `<li>` de una opción.

```
<ul id="menu">
  <li><div>Toys (n/a)</div></li>
  <li><div>Books</div></li>
  <li><div>Clothing</div></li>
  <li>
    <div>Electronics</div>
    <ul>
      <li><div>Home Entertainment</div></li>
      <li><div>Car Hifi</div></li>
      <li><div>Utilities</div></li>
    </ul>
  </li>
</ul>
```



The screenshot shows a vertical menu with four items: 'Toys (n/a)', 'Books', 'Clothing', and 'Electronics'. The 'Electronics' item is highlighted in blue and has a sub-menu open to its right, containing three items: 'Home Entertainment', 'Car Hifi', and 'Utilities'.

Puede indicarse si una opción está deshabilitada y no puede por tanto seleccionarse asignando el estilo: `"ui-state-disabled"`:

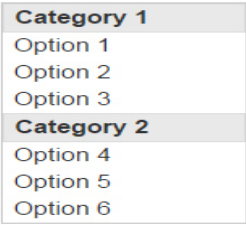
```
<ul id="menu">
  <li><div>Home Entertainment</div></li>
  <li><div>Car Hifi</div></li>
  <li><div>Utilities</div></li>
  <li class="ui-state-disabled"><div>Opcion deshabilitada</div></li>
</ul>
```



The screenshot shows a vertical menu with four items: 'Home Entertainment', 'Car Hifi', 'Utilities', and 'Opcion deshabilitada'. The 'Home Entertainment' item is highlighted in blue. The 'Opcion deshabilitada' item is shown in a lighter, disabled state.

Finalmente, también es posible mostrar una opción como una categoría de opciones asignando el estilo `"ui-widget-header"`:

```
<ul id="menu">
  <li class="ui-widget-header"><div>Category 1</div></li>
  <li><div>Option 1</div></li>
  <li><div>Option 2</div></li>
  <li><div>Option 3</div></li>
  <li class="ui-widget-header"><div>Category 2</div></li>
  <li><div>Option 4</div></li>
  <li><div>Option 5</div></li>
  <li><div>Option 6</div></li>
</ul>
```



The screenshot shows a vertical menu with six items. The first item, 'Category 1', is highlighted in a grey header style. It is followed by three options: 'Option 1', 'Option 2', and 'Option 3'. The fourth item, 'Category 2', is also highlighted in a grey header style. It is followed by three options: 'Option 4', 'Option 5', and 'Option 6'.

Cuando el usuario selecciona una opción del menú puede obtenerse la referencia al elemento `<li>` seleccionado. Para ello debe llamarse a la función **`menu()`** indicando como argumento un objeto con el atributo **`select`** asignado a una función de retrollamada que se ejecutará cuando el usuario selecciona un elemento.

```
<script>
$(function () {
    // Asignacion de la lista id="menu" como menu.
    $("#menu").menu({
        // Funcion de retrollamada al seleccionarse un elemento del menú.
        select: function (event, ui) {
            alert(ui.item.text()); // Muestra el texto de la opcion seleccionada
        }
    });
});
</script>
```

## ToolTip

<http://jqueryui.com/tooltip/>

Este componente mejora el efecto del atributo **`title`** de HTML5 que permite mostrar un cuadro informativo encima de un elemento al dejar el puntero del ratón encima.

ToolTip

Aqui hay un tooltip

Para ello se hace referencia a un párrafo o al propio documento, se llama a la función **`tooltip()`**, y todos los elementos con el atributo **`"title"`** pasan a convertirse widgets.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title></title>
    <!-- Tema 'base' para JQueryUI -->
    <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
    <!-- Libreria JQuery 3.1.0 comprimida -->
    <script src="https://code.jquery.com/jquery-3.1.0.min.js"
            integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtlJj06wt/s="
            crossorigin="anonymous"></script>
    <!-- Libreria JQueryUI 1.12.0 comprimida -->
    <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
            integrity="sha256-eGE6blurk5shJ+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
            crossorigin="anonymous"></script>

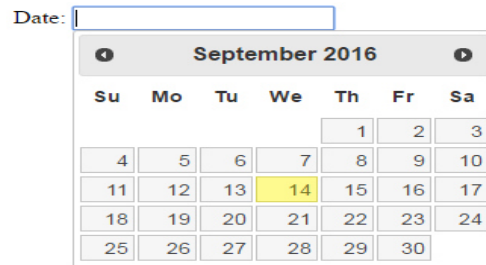
    <script>
        $( function() {
            $( document ).tooltip();
        });
    </script>

    <style>
        /* Estilo de las etiquetas */
        label {
            display: inline-block;
            width: 5em;
        }
    </style>
</head>
<body>
    <p><a href="#" title="Aqui hay un tooltip">ToolTip</a></p>
</body>
</html>
```



**Selector de fechas ( *DatePicker* )**

Este widget permite mostrar una pequeña ventana con un calendario que facilita la selección de una fecha en una caja de texto:



*Widget selector de fechas con el estilo básico*

(\*) Esta funcionalidad está presente en el navegador Google Chrome al emplear un elemento `<input type="date">`, pero no en el resto de navegadores. Este widget funciona con todos los navegadores.

Este widget se crea a partir de un simple elemento `<input type="text">`. Este elemento debe ser seleccionado desde Javascript e invocar la función ***datepicker()*** sobre él.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title></title>
  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtiJj06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurk5shj+rmkfsGyEkyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>
  <script>
    $( function() {
      $( "#datepicker" ).datepicker();
    } );
  </script>
</head>
<body>
  <p>Date: <input type="text" id="datepicker"></p>
</body>
</html>
```

El formato con el que la fecha seleccionada queda indicada en la caja de texto resultante puede indicarse añadiendo una segunda llamada a la función ***datepicker()***:

```
<script>
  $(function () {
    $("#datepicker").datepicker();
    $("#datepicker").datepicker("option", "dateFormat", "dd/mm/yy");
  });
</script>
```

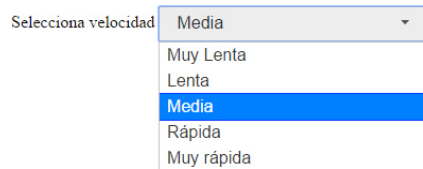
Existen más ejemplos de personalización para este widget en la web oficial:

<http://jqueryui.com/datepicker/#default>

## Lista desplegable ( *Selectmenu* )

<https://jqueryui.com/selectmenu/>

Este componente mejora las listas de selección de opciones del elemento `<select>` de HTML5 permitiendo más opciones de personalización.



Widget lista desplegable con tema básico

El widget se crea a partir de un elemento `<select>`, el cual debe seleccionarse y llamar a la función ***selectmenu()***:

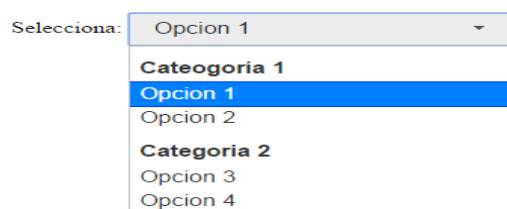
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title></title>

  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtlJj06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurk5sHj+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>

  <script>
    $(function () { $("#velocidad").selectmenu(); } );
  </script>
</head>
<body>
  <label for="velocidad">Elige una velocidad</label>

  <select name="velocidad" id="velocidad">
    <option>Muy lenta</option>
    <option>Lenta</option>
    <option selected="selected">Media</option>
    <option>Rápida</option>
    <option>Muy rápida</option>
  </select>
</body>
</html>
```

Pueden añadirse elementos no seleccionables que determinen una categoría entre las opciones mostradas incluyendo elementos `<optgroup>` y sus opciones correspondientes anidadas:



Widget lista desplegable con categorías en tema básico

```

<script>
    $(function () {
        $("#seleccion").selectmenu();
    });
</script>
</head>
<body>
    <label for="seleccion">Selecciona:</label>
    <select name="seleccion" id="seleccion">
        <optgroup label="Categoría 1">
            <option value="opc1">Opcion 1</option>
            <option value="opc2">Opcion 2</option>
        </optgroup>
        <optgroup label="Categoría 2">
            <option value="opc3">Opcion 3</option>
            <option value="opc4">Opcion 4</option>
        </optgroup>
    </select>
</body>

```

Puede hacerse que la lista se despliegue hasta una determinada altura mostrando el resto de opciones mediante una barra de desplazamiento.



Widget lista desplegable con barra de desplazamiento y tema básico

Para ello se llama dos veces a la función **selectmenu()** indicando el valor "menuWidget".

```

$("#number")
    .selectmenu()
    .selectmenu("menuWidget")
    .addClass("overflow");

```

Para determinar la altura máxima de la lista al desplegarse debe añadirse un estilo, en este caso; "overflow":

```

<style>
    .overflow{
        height: 200px;
    }
</style>

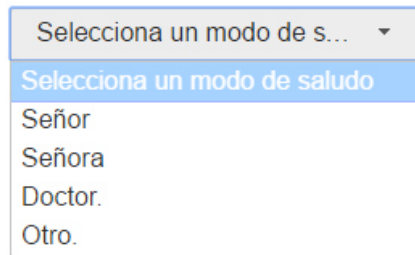
```

Para marcar una opción predeterminada que muestre un mensaje informativo puede añadirse los atributos **disable selected** en el elemento **<option>** primero.

```

<label for="saludo">Indica la forma de saludo</label>
<select name="saludo" id="saludo">
    <option disabled selected>Selecciona un modo de saludo</option>
    <option>Señor</option>
    <option>Señora</option>
    <option>Doctor.</option>
    <option>Otro.</option>
</select>

```



Para obtener el elemento seleccionado puede pasarse a la función ***selectmenu()*** un objeto como argumento con el atributo ***change*** asociado a una función de retrollamada que se ejecutará cada vez que el usuario seleccione un elemento.

Esta función recibe como segundo parámetro (*data*) la referencia al elemento `<option>` seleccionado por el usuario:

```
<script>
  $(function () {
    $("#saludo").selectmenu({
      change: function (event, data) {
        // Muestra el valor del atributo "value" del <option> seleccionado
        alert( data.item.value );
      }
    });
  })
</script>
```

## Slider

<http://jqueryui.com/slider/>

Este widget muestra una barra provista de un marcador que puede desplazarse a lo largo de la misma para determinar un valor numérico entre un máximo y un mínimo.



El componente se crea a partir de una caa que se selecciona desde javascript y se llama a la función ***slider()***. La función acepta un objeto como argumento cuyos atributos determinan la configuración del widget:

- ***max*** → Valor máximo
- ***min*** → Valor mínimo
- ***value*** → Valor inicial
- ***slide*** → función de retrollamada que se ejecuta cada vez que se cambiar de valor.

**Ejemplo:** El siguiente código muestra una página con un slider que al desplazarse permite seleccionar un valor comprendido entre 0 y 255 que es automáticamente mostrado en un párrafo:



116

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title></title>

  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtlJj06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurk5sHj+rmkfsGyEkyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>

  <script>
    $( function() {
      $("#slider").slider({
        max: 255,
        value: 127,
        min: 0,
        slide: function (event, ui) {
          $("#valor").text(ui.value); // Muestra en el párrafo el valor.
        }
      });
    } );
  </script>
</head>
<body>
  <div id="slider"></div><br />
  <p id="valor"></p>
</body>
</html>
```

Para obtener o modificar el valor seleccionado en el widget sólo es necesario llamar a la función **slider()** indicando como argumento el valor “value”.

**Ejemplo:** El siguiente código muestra un slider junto con un botón y un párrafo en el que se muestra el valor seleccionado en el widget al pulsar el botón.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title></title>

  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtlJj06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurk5sHj+rmkfsGyEkyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>

  <script>
    $( function() {
      var widget_slider = $("#slider").slider();
      $("#btnContar").on("click", function () { // Al pulsar el boton
        $("#cantidad").text(widget_slider.slider("value")); // Mostrar el valor en el
      }); // párrafo.
    });
  </script>
```

```

</script>
</head>
<body>
  <div id="slider"></div>
  <p id="cantidad"></p>
  <button id="btnContar">MOSTRAR</button>
</body>
</html>

```

Para asignar un valor al spinner se llama a la función **slider()** indicando como argumento un objeto con el atributo "value" y el valor.

```

$("#btnAsignar").on("click", function () {
  widget_slider.slider("value", 20 }); // Asigna el valor 20 al widget
})

```

## Spinner

<https://jqueryui.com/spinner/>

Este componente representa una versión más avanzada de la caja de texto **<input type="number">** pero permite personalizarla para admitir diferentes tipos de valores tales como monedas, decimal, posiciones geográficas, tiempos... etc.



Este componente se basa en un elemento **<input>** al que se hace referencia desde Javascript invocando la función **spinner()**:

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title></title>
  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtlJj06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurkSsHj+rmkfsGyEkyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>
  <script>
    $( function() {
      $("#spinner").spinner();
    });
  </script>
</head>
<body>
  <p>
    <label for="spinner">Select a value:</label>
    <input id="spinner" name="value">
  </p>
</body>
</html>

```

El widget puede personalizarse llamando a la función ***spinner()*** y pasando como argumento un objeto con diferentes atributos:

- ***min*** → Valor mínimo permitido
- ***max*** → Valor máximo permitido
- ***step*** → Valor de paso al pulsar los botones de incremento/decremento.
- ***start*** → Valor inicial

**Ejemplo:** El siguiente código muestra un control spinner que admite valores comprendidos entre 5 y 2500, con paso de 25 y un valor inicial de 1000. Se muestran además dos botones que permiten obtener su valor y asignar el valor 100.

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title></title>

  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU58gtlj06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurk5sHj+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>

  <script>
    $(function () {
      var widget_spinner = $("#spinner").spinner(
        {
          min: 5,
          max: 2500,
          step: 25,
          start: 1000
        }
      );

      $("#btnAsignar").on("click", function () {
        widget_spinner.spinner("value", 555); // Asigna valor al spinner
      })
      $("#btnMostrar").on("click", function () {
        alert(widget_spinner.spinner("value")); // Muestra valor al spinner
      })
    });
  </script>
</head>
<body>
  <p>
    <label for="spinner">Select a value:</label>
    <input id="spinner" name="value">
  </p>
  <button id="btnAsignar">ASIGNAR</button>
  <button id="btnMostrar">MOSTRAR</button>
</body>
</html>

```

## Efectos

<http://jqueryui.com/effect/>

jQueryUI implementa una serie de efectos visuales que pueden aplicarse a los elementos de una página. Estos efectos incluyen:

- **Transiciones de estilos** → Al agregar estilos a un elemento efecto visual se aplica mediante una transición en vez de inmediatamente.
- **Transición de colores** → Genera transiciones entre estilos con diferentes colores.
- **Transición de visibilidad** → Extiende los efectos de ocultación y visualización de elementos de JQuery añadiendo más efectos y animaciones.

### Efectos de transición de visibilidad. `$.hide()` / `$.show()`

jQueryUI implementa una gran cantidad de efectos de transición de visibilidad que pueden emplearse para personalizar el modo en que los elementos de la página se muestran y ocultan. Para ello se emplean las funciones `$.hide()` y `$.show()` ya vistas en JQuery, pero que son extendidas por JQueryUI.

Ambas funciones se emplean reciben el mismo conjunto de parámetros:

`$.show( <efecto>, <opciones>, <duracion_transicion>, <función_retrollamada> )`

`$.hide( <efecto>, <opciones>, <duración_transicion>, <función_retrollamada> )`

- **efecto** → Es una cadena que determina el efecto a aplicar:
  - **blind** → Oculta el elemento como una cortinilla hacia arriba
  - **bounce** → El elemento se sacude en el eje vertical
  - **clip** → Oculta el elemento en efecto cortinilla hacia arriba y abajo.
  - **drop** → El elemento desaparece ocultándose por el borde izquierdo.
  - **explode** → El elemento desaparece fragmentándose.
  - **fade** → El elemento se oculta desvaneciéndose.
  - **fold** → El elemento se oculta recogándose de abajo a arriba y hacia la izquierda.
  - **highlight** → El elemento se destaca modificando un instante su color de fondo
  - **puff** → El elemento se desvanece creciendo de tamaño.
  - **pulsate** → El elemento parpadea.
  - **scale** → El elemento se escala de su tamaño actual a otro.
  - **Shake** → El elemento se sacude en el eje horizontal.
  - **Size** → El elemento cambia de tamaño
  - **Slide** → El elemento reaparece saliendo del lado izquierdo.
  - **Transfer** → El elemento figura moverse a otra posición.



## CURSO DE PROGRAMACION JAVASCRIPT

- **opciones** → Objeto opcional de configuración para el efecto. Aunque este parámetro es opcional, algunos efectos requieren parámetros de configuración para funcionar como es el caso de “size” o “transfer”.
- **duración** → Duracion del efecto expresado en milisegundos ( opcional. Valor por defecto 400 msgs )
- **función\_retrollamada** → Funcion opcional para ejecutarse al completarse la animación del efecto.

(\*) El efecto visual de cada uno de los efectos y sus parámetros requeridos de configuración pueden probarse en la web oficial de *jQueryUI*.

**Ejemplo:** El siguiente código muestra con página provista de una capa con contenido y dos botones “APARECER” y “DESAPARECER” que muestran y ocultan respectivamente la capa de contenido al ser pulsados.

En este caso se emplea el efecto “**explode**” con las funciones **\$.hide()** y **\$.show()** indicando una duracion de 1 segundo para ambos efectos, tras lo cual se invoca la función de retrollamada **terminado()**.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtiJj06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurk5sHj+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>
  <script>
    // Función de retrollamada al terminar el efecto.
    function terminado() {
      alert("TERMINADO");
    }
    $(function () {
      // Pulsacion de boton de mostrado ( show ).
      $("#btnAparece").on("click", function () {
        $("#contenido").show("explode", {}, 1000, terminado);
      });
      // Pulsacion de boton de ocultado ( hide ).
      $("#btnDesaparece").on("click", function () {
        $("#contenido").hide("explode", {}, 1000, terminado);
      });
    });
  </script>
</head>
<body>
  <div>
    <div id="contenido" class="ui-widget-content ui-corner-all">
      <h3 class="ui-widget-header ui-corner-all">Effect</h3>
      <p>
        Etiam libero neque, luctus a, eleifend nec, semper at, lorem. Sed pede. Nulla
        lorem metus, adipiscing ut, luctus sed, hendrerit vitae, mi.
      </p>
    </div>
  </div>
  <button id="btnAparece">APARECER</button>
  <button id="btnDesaparece">DESAPARECER</button>
</body>
</html>
```

## Efectos de transición `$.effect()`

jQueryUI incluye la función `$.effect()` que puede emplearse para lanzar ciertos efectos que permiten destacar un elemento, modificando su tamaño o posición. Esta función acepta los mismos parámetros que las funciones `$.show()` y `$.hide()`:

**Ejemplo:** La siguiente página muestra una capa con contenido que se sacude con los efectos “*bounce*” y “*shake*” al pulsar los botones mostrados:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title></title>
  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtiJj06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurk5sHj+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>

  <script>
    // Función de retrollamada al terminar el efecto.
    function terminado() {
      alert("TERMINADO");
    }
    $(function () {
      $("#btnEfBounce").on("click", function () {
        $("#contenido").effect("bounce", {}, 1000, terminado);
      });
      $("#btnShake").on("click", function () {
        $("#contenido").effect("shake", {}, 1000, terminado);
      });
    });
  </script>
</head>
<body>
  <div>
    <div id="contenido" class="ui-widget-content ui-corner-all">
      <h3 class="ui-widget-header ui-corner-all">Effect</h3>
      <p>
        Etiam libero neque, luctus a, eleifend nec, semper at, lorem. Sed pede. Nulla
        lorem metus, adipiscing ut, luctus sed, hendrerit vitae, mi.
      </p>
    </div>
  </div>
  <button id="btnEfBounce">BOUNCE</button>
  <button id="btnShake">SHAKE</button>
</body>
</html>
```

## Transición de estilos `$.addClass()`, `$.removeClass()`, `$.toggleClass()`

jQueryUI permite crear animaciones del cambio de un elemento al aplicarle diferentes estilos mediante las funciones `$().addClass()`, `$().removeClass()`, `$().toggleClass()`.

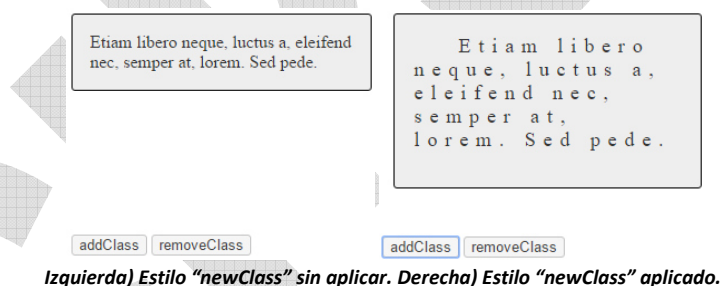
```
$.addClass(<clase estilo>, <duración transición>, <función terminación>);
```

```
$.removeClass(<clase estilo>, <duración transición>, <función terminación>);
```

La función `$(selector).addClass(clase)` permite añadir una clase de estilo a un elemento generando una animación de transición entre el aspecto actual y el aspecto resultante al añadir la clase indicada en el parámetro `<clase_estilo>`. La duración de la transición viene expresada en milisegundos en el parámetro `<duracion>`. Opcionalmente puede indicarse el nombre de una función de retrollamada como parámetro `<función>` para ser ejecutada al completarse la transición.

La función `$().removeClass()` hace el proceso inverso generando una animación de transición entre el aspecto actual y el resultante al eliminar la clase de estilo indicada. Los parámetros aceptados son los mismos.

**Ejemplo:** La siguiente página muestra una capa a la que se le añade y elimina una clase de estilo ( *newClass* ) cuando se pulsán dos botones “*addClass*” y “*removeClass*” situados en la parte inferior:



Al pulsarse los botones, se produce una transición entre ambos aspectos de un segundo de duración.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title></title>
  <style>
    .toggler {
      width: 500px;
      height: 200px;
      position: relative;
    }
    #button {
      padding: .5em 1em;
      text-decoration: none;
    }
    #effect {
      width: 240px;
      padding: 1em;
      border: 1px solid #000;
```

```

        background: #eee;
        color: #333;
    }
    .newClass {
        text-indent: 40px;
        letter-spacing: .4em;
        width: 410px;
        height: 120px;
        padding: 30px;
        margin: 10px;
        font-size: 1.1em;
    }
</style>

<!-- Tema 'base' para JQueryUI -->
<link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet"
/>

<!-- Libreria JQuery 3.1.0 comprimida -->
<script src="https://code.jquery.com/jquery-3.1.0.min.js"
        integrity="sha256-cCueBR6CsyA4/9sZpPfrX3s49M9vUU5BgtlJj06wt/s="
        crossorigin="anonymous"></script>
<!-- Libreria JQueryUI 1.12.0 comprimida -->
<script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
        integrity="sha256-eGE6blurk5sHj+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
        crossorigin="anonymous"></script>

<script>
    $(function () {
        $("#button_add").on("click", function () {
            // Añado estilo
            $("#effect").addClass("newClass", 1000);
        });
        $("#button_remove").on("click", function () {
            // Suprimo estilo
            $("#effect").removeClass("newClass", 1000);
        });
    });
</script>
</head>
<body>
    <div class="toggler">
        <div id="effect" class="ui-corner-all">
            Etiam libero neque, luctus a, eleifend nec, semper at, lorem. Sed pede.
        </div>
    </div>
    <!-- Botón de agregado de estilo -->
    <button id="button_add" class="ui-state-default ui-corner-all">addClass</button>
    <!-- Botón de agregado de estilo -->
    <button id="button_remove" class="ui-state-default ui-corner-all">removeClass</button>
</body>
</html>

```

### Transición de color `$.animate()`

La función `$.animate()` en JQueryUI permite además la transición de colores, de modo que es posible añadir estilos relativos a colores y ver cómo se produce una transición entre los colores actuales y los indicados.

`$.animate(<estilos>, <duración_transición>)`

Ejemplo: El siguiente código provoca una transición de 1 segundo de duración en la que se modifica el color de fondo, y el color de texto de un elemento con id="effect":

```

$( "#effect" ).animate({
    backgroundColor: "#fff",
    color: "#000"
}, 1000 );

```

## Interacciones

Las interacciones son versiones avanzadas de los eventos clásicos que pueden detectarse JQueryUI y que representan modelos de interaccion complejos tales como:

- **Selectable** → Elegir un elemento
- **Draggable** → Arrastrar un elemento a otra posicion de la página
- **Droppable** → Arrastrar un elemento y soltarlo dentro de otro
- **Resizable** → Modificar el tamaño de un elemento mediante un “tirador” que se muestre en el margen inferior izquierdo

### Selección de elementos `$(...).selectable()`

<http://jqueryui.com/selectable/>

Este modelo de interaccion permite seleccionar uno o varios elementos contenidos en un elemento lista `<ol>`. Al hacer clic sobre cada uno de ellos quedan seleccionados y puede obtenerse la referencia de todos ellos.

Es importante destacar que implementar la selección de elementos emplea ciertas clases de estilos que debemos recordar para poder definir los colores y aspecto de los elementos según su estado:

- **.ui-selecting** → Estilo de un elemento cuando el usuario hacer clic sobre él.
- **.ui-selected** → Estilo de un elemento seleccionado.

Para hacer que la lista de elementos sea seleccionable debe obtenerse la referencia al elemento y llamar a la función `$(...).selectable()`.

**Ejemplo:** la siguiente página muestra una lista de elementos que el usuario puede seleccionar. Al hacerlo se muestra en un párrafo superior los índices de los elementos seleccionados:

You've selected: #1 #3 #4.

Item 1
Item 2
Item 3
Item 4
Item 5
Item 6

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style>
    #selectable .ui-selecting { background: #FECA40; }
    #selectable .ui-selected { background: #F39814; color: white; }
    #selectable { list-style-type: none; margin: 0; padding: 0; width: 60%; }
    #selectable li { margin: 3px; padding: 0.4em; font-size: 1.4em; height: 18px; }
  </style>
  <!-- Tema 'base' para JQueryUI -->
  <link href="https://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css" rel="stylesheet" />
  <!-- Libreria JQuery 3.1.0 comprimida -->
  <script src="https://code.jquery.com/jquery-3.1.0.min.js"
    integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU58gtiJj06wt/s="
    crossorigin="anonymous"></script>
  <!-- Libreria JQueryUI 1.12.0 comprimida -->
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"
    integrity="sha256-eGE6blurk5sHj+rmkfsGYeKyZx3M4bG+ZlFyA7Kns7E="
    crossorigin="anonymous"></script>
  <script>
    $(function () {
      // La lista es declarada como seleccionable
      $("#selectable").selectable();
    });
  </script>
</head>
<body>
<p id="feedback">
  <span>You've selected:</span> <span id="select-result">none</span>.
</p>
  <!-- Lista de opciones seleccionables -->
  <ol id="selectable">
    <li class="ui-widget-content">Item 1</li>
    <li class="ui-widget-content">Item 2</li>
    <li class="ui-widget-content">Item 3</li>
    <li class="ui-widget-content">Item 4</li>
    <li class="ui-widget-content">Item 5</li>
    <li class="ui-widget-content">Item 6</li>
  </ol>
</body>
</html>

```

La función `$(selector).selectable()` puede invocarse pasando como argumento opcional un objeto con un atributo **“stop”** y una función de retollamada que se ejecute al hacer el usuario sobre un elemento de la lista. Esta función puede obtener los elementos seleccionados seleccionando aquellos que tengan la clase de estilo **“.ui-selected”**:

```

<script>
  $(function () {
    // La lista es declarada como seleccionable
    $("#selectable").selectable({
      // Funcion de retollamada para cada seleccion
      stop: function () {
        // Párrafo de salida
        var result = $("#select-result").empty();
        // Seleccion de los elementos con el estilo .ui-selected aplicado
        $(".ui-selected", this).each(function () {
          // Obtencion de índice de elemento seleccionado
          var index = $("#selectable li").index(this);
          // Agregado a cada de texto
          result.append( " #" + ( index + 1 ) );
        });
      }
    });
  });
</script>

```