



**UTN.BA** FACULTAD  
REGIONAL  
BUENOS AIRES  
SECRETARÍA DE EXTENSIÓN UNIVERSITARIA FRBA UTN

**Centro de  
e-Learning**

# Javascript, JQuery y JSON avanzado.



[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)



## Módulo 1

# INTRODUCCIÓN A JQUERY



**Creación de menú desplegable, sliders,  
galeria de fotos y reproductor de audio  
haciendo uso de las herramientas  
brindadas por JQuery.**



## **Presentación de la Unidad:**

**Haciendo uso de las herramientas vistas en la Unidad I aprenderemos a crear un menú desplegable, un slider, una galería de fotos y un reproductor de audio.**

**Veremos como con la utilización de simples métodos de JQuery podemos desarrollar atractivos sliders o galerías de fotos.**



## Objetivos:

- ❖ **Hacer uso de los métodos para manipulación de css y para modificación de atributos HTML.**
- ❖ **Poner en práctica los selectores, eventos y efectos que nos brinda JQuery en ejemplos sencillos que resultarán de gran utilidad.**



## Temario:

**Ejemplo 1: Desplegable**

**Ejemplo 2: Slider versión 1**

**Ejemplo 3: Slider versión 2**

**Ejemplo 4: Slider versión 3**

**Ejemplo 5: Galería de fotos**

**Ejemplo 6: Reproductor de audio**



## CONSIGNAS PARA EL APRENDIZAJE COLABORATIVO

En esta Unidad los participantes se encontrarán con diferentes tipos de consignas que, en el marco de los fundamentos del MEC\*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:



1. Los foros asociados a cada una de las unidades.
2. La Web 2.0.
3. Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen las actividades sugeridas y compartan en los foros los resultados obtenidos.



En esta unidad desarrollaremos una serie de ejemplos prácticos en los cuales veremos como aplicando las herramientas vistas en la anterior unidad de manera sencilla logramos un gran atractivo estético.

## EJEMPLO 1

Para hacer este menú desplegable hemos utilizado los eventos `mouseenter`, para indicar las acciones a realizar una vez que el mouse entra en el ítem de la lista y `mouseleave`, para indicar las acciones una vez que el mouse abandona el elemento. Y hemos utilizado los efectos de `slideDown` y `slideUp`.

El menú está hecho con CSS, de manera que la `<ul>` principal aparece visible y las `<ul>` que están comprendidas dentro de los `<li>` superiores aparecen con `display none`.

Este mismo menú podría trabajarse solo con CSS definiendo para la pseudoclase `:hover` del menú principal `display: block` de la lista desplegable de la siguiente manera:

```
ul.menu li:hover ul {  
    display: block;  
}
```

En lugar de ello hemos asignado un id a cada una de las `<ul>` desplegables, `desplegable1` y `desplegable2` y un id a cada uno de los `<li>` de la `<ul>` principal:

```
<ul class="menu">  
<li id="opcion1"><a href="#principal">Principal</a></li>  
<li id="opcion2"><a href="#ropa">Ropa</a>  
    <ul id="desplegable2">  
        <li><a href="#">Deportiva</a></li>  
        <li><a href="#">Vestir</a></li>  
        <li><a href="#">Fiesta</a></li>  
    </ul>  
</li>  
<li id="opcion3"><a href="#accesorios">Accesorios</a>  
    <ul id="desplegable3">  
        <li><a href="#">Zapatos</a></li>  
        <li><a href="#">Carteras</a></li>  
        <li><a href="#">Bijoux</a></li>  
    </ul>  
</li>  
<li id="opcion4"><a href="#contacto">Contacto</a></li>  
</ul>
```



Y hemos definido el evento `mouseenter()` sobre los `<li>` cuyo id son `opcion2` y `opcion3` ante el cual se lleva a cabo el efecto `slideDown` sobre `#desplegable2` y `#desplegable3` respectivamente. Y el evento `mouseleave()` sobre los `<li>` cuyo id son `opcion2` y `opcion3` ante el cual se lleva a cabo el efecto `slideUp` sobre `#desplegable2` y `#desplegable3` respectivamente.

```
$(document).ready(function(){  
  $("#opcion2").mouseleave(function(event){  
    event.preventDefault();  
    $("#desplegable2").slideUp();  
  });
```

```
  $("#opcion2").mouseenter(function(event){  
    event.preventDefault();  
    $("#desplegable2").slideDown();  
  });
```

```
  $("#opcion3").mouseleave(function(event){  
    event.preventDefault();  
    $("#desplegable3").slideUp();  
  });
```

```
  $("#opcion3").mouseenter(function(event){  
    event.preventDefault();  
    $("#desplegable3").slideDown();  
  });
```

```
});
```

El código completo de este ejemplo se encuentra en la carpeta desplegable.

## EJEMPLO 2

Vamos a desarrollar un slider haciendo uso de algunos de los métodos vistos. El código del ejemplo es el siguiente:

Slider1.html

```
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
<title>Documento sin título</title>  
<script type="text/javascript" src="jquery-1.10.2.min.js"></script>  
<script type="text/javascript">
```

```
$(document).ready(function(){
    function animacion(){
        $("#foto1").fadeIn(2000, function(){
            $("#foto1").fadeOut(2000, function(){
                $("#foto2").fadeIn(2000, function(){
                    $("#foto2").fadeOut(2000, function(){
                        $("#foto3").fadeIn(2000,
function(){$("#foto3").fadeOut(2000,
function(){animacion();
}}}}}}}}));
    }
    animacion();
});
</script>
<style type="text/css">
    #imagenes{
        width: 600px;
        height: 450px;
        border: 1px solid grey;
        position: relative;
    }
    #foto1, #foto2, #foto3{
        display: none;
        position: absolute;
    }
</style>
</head>

<body>
<h2>Slider</h2>
<div id="imagenes">
    
    
    
</div>
</body>
</html>
```

Como podrán observar lo que tenemos en el body de nuestro html es un div con un id="imagenes" en el cual tenemos tres etiquetas de imágenes a las que le asignamos un id, foto1, foto2 y foto3 respectivamente para poder tomarlas desde JQuery mediante ese id.

Como podrán observar en el css de la página las imágenes tienen position: absolute; esto es para que las tres imágenes se posicionen una encima de la otra y el div tiene position: relative para que el posicionamiento absoluto de las imágenes sea tomado desde el div que las contiene y no desde la esquina superior izquierda de la pantalla.

Luego en el código JQuery como verán lo que se hace es un callback de funciones apilando los efectos que le aplicaremos a las imágenes. Como inicialmente las imágenes no se visualizan ya que todas están con display: none; en nuestro css lo que hacemos es aplicar el efecto fadeIn() a la imagen cuyo id es foto1 en 2000 milisegundos y luego, una vez finalizado el mismo, aplicar el efecto fadeOut() sobre la imagen con id foto1, una vez finalizado aplicamos fadeIn() sobre la foto2 y luego fadeOut() sobre la foto2 y por último hacemos lo mismo con la foto3.

Esto sería el código logrado a través del callback de funciones siguiente:

```
$("#foto1").fadeIn(2000, function(){  
    $("#foto1").fadeOut(2000, function(){  
        $("#foto2").fadeIn(2000, function(){  
            $("#foto2").fadeOut(2000, function(){  
                $("#foto3").fadeIn(2000, function(){  
                    $("#foto3").fadeOut(2000, function(){  
                        }  
                    }  
                }  
            }  
        }  
    }  
});
```

Pero si nosotros utilizáramos solamente este código, esta pila de efectos se ejecutaría una sola vez y finalizaría, es decir se irían sucediendo el fadeIn y fadeOut de cada imagen una sola vez y luego veríamos el recuadro representado por nuestro div vacío. Entonces lo que hacemos para que estos efectos se ejecuten indefinidamente y una vez que termine toda la secuencia comience nuevamente es utilizar una función recursiva. El concepto de función recursiva es el de una función que se llama así misma dentro de la misma función antes de finalizar, y esto hace que la misma se ejecute indefinidamente.

Entonces dentro de nuestro \$(document).ready() la llamamos desde fuera de la función para que se ejecute por primera vez y luego dentro de la misma función volvemos a hacer el llamado a sí misma y esto es lo que hará que los efectos se ejecuten indefinidamente.

## EJEMPLO 3

Haremos una pequeña modificación al ejemplo anterior manteniendo la estructura del mismo pero modificando el efecto para lograr una variante del slider.

Slider2.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar



```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin título</title>
<script type="text/javascript" src="jquery-1.10.2.min.js"></script>
<script type="text/javascript">

    $(document).ready(function(){

        function animacion(){
            $("#foto1").animate({"left": "+=600px"}, 4000, function(){
                $("#foto2").animate({"left": "+=600px"}, 4000, function(){
                    $("#foto3").animate({"left": "+=600px"}, 4000,
function(){

                                $("#foto1").css("left", "0")
                                $("#foto2").css("left", "0")
                                $("#foto3").css("left", "0")

                                animacion();

                            }}}});
        }
        animacion();

    });
</script>
<style type="text/css">
    #imagenes{
        width: 600px;
        height: 450px;
        border: 1px solid grey;
        position: relative;
        overflow: hidden;
        background-image: url(foto1.jpg)
    }
    #foto1, #foto2, #foto3{
        display: block;
        position: absolute;
    }

</style>
</head>

<body>
<h2>Slider</h2>
<div id="imagenes">
    
</div>
</body>
</html>
```

```



</div>
</body>
</html>
```

Como verán hemos introducido algunas modificaciones en nuestro código JQuery y en el css para poder lograr esta variante. Veamos cuales son esas modificaciones y cual es el objetivo de cada una.

Ahora nuestras imágenes ya no se irán desvaneciendo sino que queremos que vayan deslizándose una sobre la otra. Para poder hacer esto utilizamos el efecto `animate()` de JQuery:

```
$("#foto1").animate({"left": "+=600px"}, 4000, function(){
```

Lo que hacemos entonces es desplazar las imágenes una a continuación de la otra en el tiempo de 4000 milisegundos. Una vez que las tres se desplazaron 600px a la izquierda volvemos a las tres a la posición 0 y comenzamos nuevamente el desplazamiento a la izquierda.

Para que al desplazar la última imagen no se vea el fondo vacío del div es que modificamos nuestro css agregándole a nuestro div `id="imágenes"` el atributo `background-image: url(foto1.jpg)` y para que las imágenes no estén visibles a la izquierda del div es que le agregamos el atributo `overflow: hidden;`

También modificamos el atributo `display: block;` en `#foto1`, `#foto2` y `#foto3` ya que ahora necesitamos que nuestras imágenes estén visibles todo el tiempo en nuestra animación.

## EJEMPLO 4

Ahora le haremos una pequeña incorporación al ejemplo generando una alternativa mas. Le agregamos un div por cada imagen que tenemos dentro del `<div id="imágenes">` y nos quedan tres divs con `id #texto1`, `#texto2` y `#texto3` a los cuales les hemos definido sus atributos de CSS y que haremos aparecer mediante el efecto `slideDown` junto con la aparición de la imagen correspondiente y desaparecer una vez que la foto desaparece.

Slider3.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // [e-learning@sceu.frba.utn.edu.ar](mailto:e-learning@sceu.frba.utn.edu.ar)

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin título</title>
<script type="text/javascript" src="jquery-1.10.2.min.js"></script>
<script type="text/javascript">

    $(document).ready(function(){

        function animacion(){
            $("#texto3").slideDown(4000, function(){
                $("#texto3").css("display", "none")
                $("#texto2").slideDown(4000, function(){
                    $("#texto2").css("display", "none")
                    $("#texto1").slideDown(4000, function(){
                        $("#texto1").css("display", "none")

                    });
                });
            });

            $("#foto3").animate({"left": "+=600px"}, 4000, function(){
                $("#foto2").animate({"left": "+=600px"}, 4000, function(){
                    $("#foto1").animate({"left": "+=600px"}, 4000,

function(){

                    $("#foto3").css("left", "0")
                    $("#foto2").css("left", "0")
                    $("#foto1").css("left", "0")

                    animacion();

                });
            });
        }
        animacion();

    });
</script>
<style type="text/css">
    #imagenes{
        width: 600px;
        height: 450px;
        border: 1px solid grey;
        position: relative;
        overflow: hidden;
        background-image: url(foto1.jpg)
    }
    #foto1, #foto2, #foto3{
        display: block;
        position: absolute;
    }
    #texto1, #texto2, #texto3{
```

```
        display: none;
        position: absolute;
        text-align: center;
        width: 600px;
        height: 30px;
        padding-top: 10px;
        vertical-align: bottom;
        background-color: #000000;
        color: #FFFFFF;
        opacity: 0.3;
        filter: alpha(opacity=30);
    }

</style>
</head>

<body>
<h2>Slider</h2>
<div id="imagenes">
    
    
    
    <div id="texto1">Esta es la descripcion de la foto 1</div>
    <div id="texto2">Esta es la descripcion de la foto 2</div>
    <div id="texto3">Esta es la descripcion de la foto 3</div>

</div>
</body>
</html>
```



Como verán a partir de estos tres ejemplos las posibilidades son infinitas, es cuestión de ir jugando con los recursos que nos brinda CSS y con los métodos disponibles en JQuery para lograr el objetivo deseado.

Estas son algunas de las alternativas que se nos ocurrieron a nosotros, los invito a que prueben ustedes las opciones que se les ocurran. Solo es cuestión de jugar y probar.

## EJEMPLO 5

Ahora veremos un ejemplo de Galería de fotos. De la misma manera que con el ejemplo de slider anterior, este es el comienzo, a partir del mismo podrán hacer infinitas variantes según los resultados que quieran obtener.

Por ejemplo podrían cambiar los efectos a utilizar, podrían hacer que la foto se oculte presionando en cualquier parte de la pantalla en lugar de presionando sobre la foto ampliada como ocurre en el ejemplo, podrían agregarle alguna imagen pequeña a continuación de la imagen ampliada como esta

CLOSE  o esta otra 

para cerrarla.

En fin, las posibilidades como bien decíamos son infinitas es solo cuestión de poner en práctica nuestra imaginación y nuestra creatividad y hacer uso de los métodos que nos brinda JQuery para conseguirlo.

Esta es una galería muy común, existen muchísimos plugins que trabajan con esta técnica, desde hace ya mucho tiempo incluso muchos de ellos están desarrollados en Javascript sin hacer uso de ningún framework. Por supuesto las hay también haciendo uso de las bondades de JQuery y de algunos otros frameworks como Mootools o Prototype por nombrar algunos.

Veamos el código del ejemplo:

Nuestro body está compuesto por un div id="previa" que tiene como atributo CSS display: none; lo que hace que el mismo no sea visible al iniciar nuestro html. Luego se encuentran dentro de un div id="container" todas las miniaturas a las cuales le asignamos al atributo alt, nótese que el alt coincide en parte con el nombre del archivo, por ejemplo si el archivo es foto\_1\_small.jpg el alt es foto\_1, digo esto porque luego lo utilizaremos en el código JQuery para hacer referencia a cada una de las imágenes

```
<div id="previa">
  <div>
    <img src="" id="imgFull">
  </div>
</div>
<div id="container">
  
  
  
  
  
  
</div>
```

Luego lo que hacemos desde JQuery es lo siguiente:

```
$("#container img").click(function(){
    var nombre = $(this).attr("alt") + "_big";
```

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar



```
var direccion = nombre + ".jpg";

$("#imgFull").attr('src',direccion);
$("#previa").fadeIn();

});
```

Lo primero es definir sobre la etiqueta img del div id="container" el evento click, lo que hará que todas nuestras imágenes estén alcanzadas por el mismo. Dentro de la función a ejecutarse una vez producido el evento armamos el nombre de la imagen a mostrar a través de \$(this).attr("alt"), con \$(this) hacemos referencia al objeto imagen clickeado, tomamos el atributo alt y luego le añadimos "\_big", en el ejemplo de foto\_1 el nombre nos quedaría foto\_1\_big y en el renglón siguiente le añadimos el .jpg. Entonces ante el click de la miniatura lo que hacemos es asignarle a la img id="imgFull" que está dentro del div id="previa" el source correspondiente y luego hacemos visible el div mediante el efecto fadeIn().

A continuación en estos renglones:

```
$("#imgFull").click(function(){
    $("#previa").fadeOut();
});
```

Lo que hacemos es ocultar nuevamente el div contenedor de la imagen ampliada ante el click sobre la misma mediante el efecto fadeOut().

## EJEMPLO 6

Vamos a ver por último un reproductor de música muy sencillo. El código correspondiente lo encuentran en la carpeta reproductor contenida en los ejemplos de la unidad.

Veamos el código:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin título</title>
<link rel="stylesheet" type="text/css" href="style_mini.css">
<script type="text/javascript" src="jquery-1.10.2.min.js"></script>
<script type="text/javascript">
$(document).ready(function() {
    $('#toggle').bind("click", function() {
```

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar

```

        if ($(this).attr("class") == "play")
            $(this).attr("class", "stop",
onclick=document.getElementById('mediaPlayer').play()).attr("title", "Pausar");
        else
            $(this).attr("class", "play",
onclick=document.getElementById('mediaPlayer').pause()).attr("title", "Encender");
    });

    $('.volup img').click(function(){
        document.getElementById('mediaPlayer').volume+=.2
    });

    $('.-vol img').click(function(){
        document.getElementById('mediaPlayer').volume-=.2
    });

});
</script>
</head>

<body>
<div id="radio" style="text-align:center;margin-top: 10px;height: 200px;">
    <div id="mediaplayer_style">
        <div class="play_stop">
            <div id="toggle" class="play" title="Encender"></div>

            </div>
            <div class="volup"></div>
            <div class="-vol"></div>
        </div>

        <audio id="mediaPlayer" preload="metadata">
            <source src="hey soul sister-train.mp3" type="audio/mpeg">
        </audio>
    </div>
</body>
</html>

```

La función bind() añade un handler a un evento determinado producido sobre un elemento. En este caso está respondiendo al evento click sobre el elemento con div id="toggle".

A continuación evalúa si el atributo class de dicho div es "play" en cuyo caso cambia la clase a "stop", lo que hará cambiar la imagen visible y aplicar el método play() sobre el



elemento audio con id="mediaPlayer".

Si la case es "stop" en lugar de "play" hace el proceso inverso, cambia la clase a "play" y detiene el sonido.

Ante el evento click sobre la imagen que indica subir el volumen aumenta el volumen 0.2.

Y ante el evento click sobre la imagen que indica bajar el volumen baja el volumen 0.2.

Y eso es todo este es un código muy sencillo para poder manejar archivos de audio. Les quiero hacer una observación, como pueden ver, hay código Javascript puro mezclado con el código JQuery. Pueden hacer esto sin problemas en la medida que les sea más práctico o mas sencillo en ciertos casos. Esto no trae ningún tipo de conflicto y funciona perfectamente.