Apartado 1.1. Estudio de las distintas librerías adicionales a jQuery.

JQuery UI:

JQuery UI se centra específicamente en la interfaz de usuario interactiva haciéndola más fácil de administrar. La integridad de JQuery UI se divide en tres categorías; interacciones, widgets y efectos.

JQuery UI introduce un nuevo tema de escala de grises más moderno y nuevos widgets (elementos prefabricados como botones, barras de progreso, pestañas, etc.) como el *Controlgroup* que actualiza el conjunto de botones y agrega soporte para selectmenu, o *checkboxradio* que mejora las casillas de verificación. Además, los widgets son más fáciles de personalizar gracias su opción 'clases'.

En la página web de jqueryui.com hay una amplía selección de demostraciones referentes a las tres categorías antes mencionadas. En el apartado de interacciones destaco la interacción *Selectable* (permite seleccionar varios elementos a la vez) y *Sortable* (permite reordenar los elementos de una lista). A parte de los widgets mencionados antes, destaco *Accordion* que muestra paneles desplegables. Y en cuanto a los efectos, estos introducen una interfaz más animada y fluida.

Buscando información por Internet he visto que JQuery se usa en 20.094.894 sitios web mientras que JQuery UI en 3.129.829. Esta diferencia se explica en que JQuery fue creada para satisfacer muchas necesidades, mientras que JQuery UI se centra en la interfaz de usuario y además está construida sobre JQuery, por lo tanto, es necesaria para poder trabajar con JQuery UI.

JQuery Tools:

JQuery Tools también es una biblioteca centrada en la interfaz de usuario que proporciona componentes como pestañas, superposiciones, información sobre herramientas y acordeones. Su principal característica sobre JQuery UI es que permite un tamaño de archivo más pequeño, aunque también depende de cada caso (calidad del código, combinación de muchas pestañas, superposiciones, etc.).

Una de las mejoras más importantes es que cada elemento cuneta con una interfaz de programación de aplicaciones (API) mediante el método *data*. Una vez identificado el elemento puedes interactuar con sus diferentes métodos (mover desplaza elementos hacia adelante, begin desplaza el elemento al principio de la página, etc.).

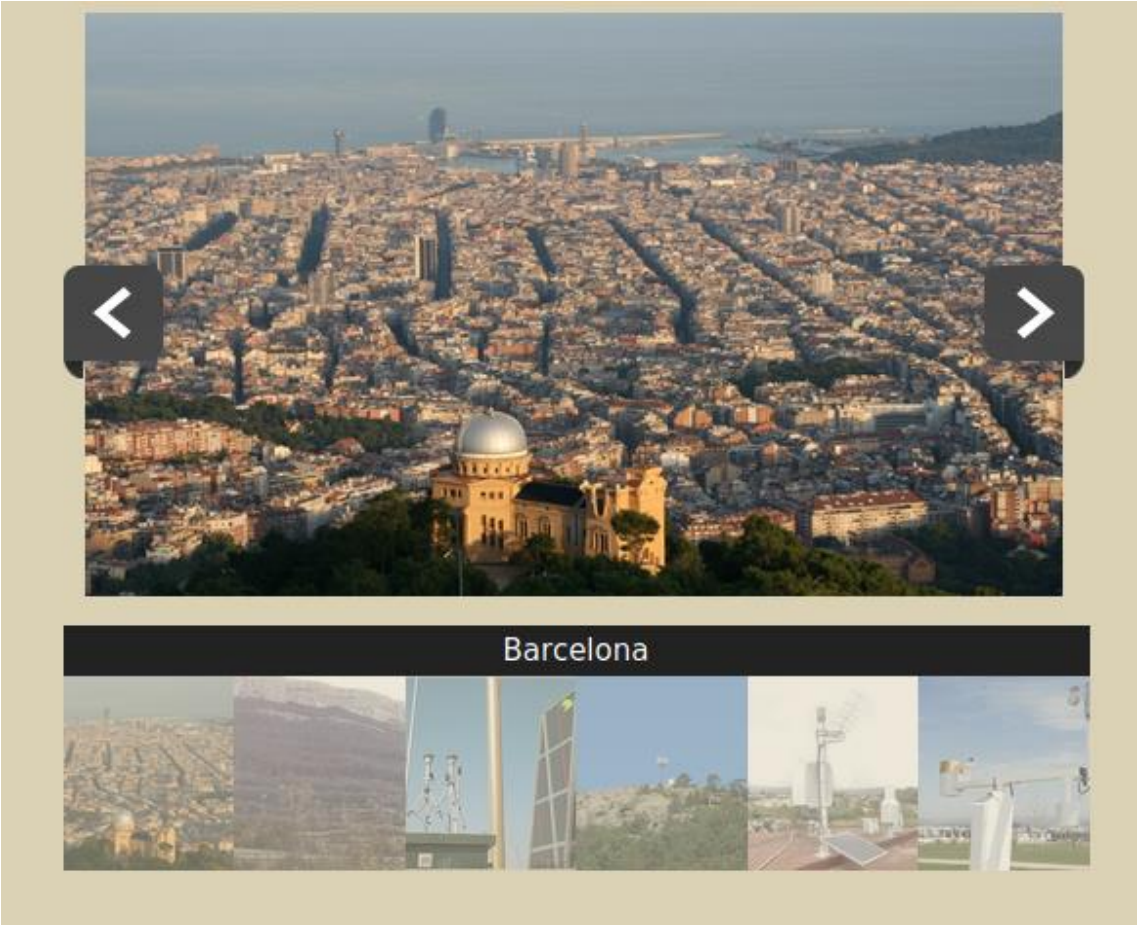
Con esta librería cada elemento cuenta con un evento *onBefore,* que ocurre antes de que se tome acción sobre la herramienta, y el evento *on* que ocurre después de la acción. Para asignar un oyente a eventos se puede usar el método *bind* o se puede suministrar un oyente desde la API. También cabe destacar la posibilidad de crear plugins.

JQuery Mobile:

Sistema de interfaz de usuario enfocado en la accesibilidad a plataformas móviles. Según he podido ver no es un sistema muy usado, y solo se recomienda si solo estás familiarizado con JQuery ya que hay mejores opciones para las plataformas móviles. Esto es debido a que JQuery Mobile no se actualiza continuamente, es más lenta que otros framework y tiene una comunidad muy pequeña.

JQuery Mobile permite la integración con JQuery para implementar las diferentes funcionalidades que contiene JQuery (como arrastrar y soltar o *hovering)* para pantallas táctiles.

Apartado 1.2. Realización de una galería de imágenes.



Todas las imágenes se encuentran “escondidas” en un contenedor, ya que tiene el overflow a hidden. La imagen principal tiene la clase ‘active’, dicha clase identificará la imagen que va a pasar a verse. Mediante los métodos addClass() y removeClass(), voy pasando la clase ‘active’ a la imagen que quiero mostrar y se la quito a la que quiero ocultar, de esta forma siempre puedo reconocer la imagen que se está mostrando. Para esconder o mostrar una imagen utilizo la propiedad ‘z-index’, que en caso de tener varios elementos superpuestos el elemento con mayor z-index cubre a los demás. Esta metodología la implemento tanto para los botones de ‘atrás’ y ‘adelante’, como para las distintas imágenes de menor tamaño desplegadas a la hora de clickarlas.

```
$(document).ready(function(){
    $('.next').on('click',function(){
        //$('.active').css('opacity', '0.5');
        var imgActiva = $('.active');
        var iNext = imgActiva.next();
        if(iNext.length == 0){
            var imgNext = $('img[alt="Barcelona"]');
        }
        else
            var imgNext = imgActiva.next();
        //var captionText = $('#caption');
        if(imgNext.length){
            imgActiva.removeClass('active').css('z-index',-10);
            imgNext.addClass('active').css('z-index',10);
            var newActiva = $('.active');
            $(".caption-container").text(newActiva.attr('alt'));
            //$('.active').css('opacity', '1');
        }
    });

    $('.prev').on('click',function(){
        //$('.active').css('opacity', '0.5');
        var imgActiva = $('.active');
        var iPrev = imgActiva.prev();
        if(iPrev.length == 0){
            var imgPrev = $('img[alt="Alicante"]');
        }
        else
            var imgPrev = imgActiva.prev();
        if(imgPrev.length){
            imgActiva.removeClass('active').css('z-index',-10);
            imgPrev.addClass('active').css('z-index',10);
            var newActiva = $('.active');
            $(".caption-container").text(newActiva.attr('alt'));
            //$('.active').css('opacity', '1');
        }
    });
```

```
$('.demoCursor').on('click',function(){
    var imgActiva = $('.active');
    //$('.demoCursor').each(function(){
    if($(this).attr('alt') == "Barcelona")
    {
        //alert($(this).attr('alt'));
        var imgElem = $('img[alt="Barcelona"]');
        imgActiva.removeClass('active').css('z-index',-10);
        imgElem.addClass('active').css('z-index',10);
        var newActiva = $('.active');
        $(".caption-container").text(newActiva.attr('alt'));
    }
    if($(this).attr('alt') == "Burgos")
    {
        //alert($(this).attr('alt'));
        var imgElem = $('img[alt="Burgos"]');
        imgActiva.removeClass('active').css('z-index',-10);
        imgElem.addClass('active').css('z-index',10);
        var newActiva = $('.active');
        $(".caption-container").text(newActiva.attr('alt'));
    }
    if($(this).attr('alt') == "Madrid")
    {
        //alert($(this).attr('alt'));
        var imgElem = $('img[alt="Madrid"]');
        imgActiva.removeClass('active').css('z-index',-10);
        imgElem.addClass('active').css('z-index',10);
        var newActiva = $('.active');
        $(".caption-container").text(newActiva.attr('alt'));
    }
    if($(this).attr('alt') == "Alcala")
    {
        //alert($(this).attr('alt'));
        var imgElem = $('img[alt="Alcala"]');
        imgActiva.removeClass('active').css('z-index',-10);
        imgElem.addClass('active').css('z-index',10);
        var newActiva = $('.active');
        $(".caption-container").text(newActiva.attr('alt'));
    }
    if($(this).attr('alt') == "Cordoba")
    {
        //alert($(this).attr('alt'));
        var imgElem = $('img[alt="Cordoba"]');
        imgActiva.removeClass('active').css('z-index',-10);
        imgElem.addClass('active').css('z-index',10);
        var newActiva = $('.active');
        $(".caption-container").text(newActiva.attr('alt'));
    }
    if($(this).attr('alt') == "Alicante")
```

La forma de que tengo de distinguir cada imagen es mediante su atributo ‘atl’, que contiene el nombre de cada ciudad mostrada. Por último, para poder recoger la galería continuamente, tanto el método next() como prev(), comprueban si hay una imagen antes o después (si su longitud es igual a 0 significa que no hay elemento) y en tal caso muestro la primera o la última imagen respectivamente.

Apartado 1.3. Muestra de tablas paginadas.

Le doy formato a las tablas justo al cargar el documento en JQuery y siguiendo una metodología parecida a la del apartado anterior (con los métodos removeClass(), addClass() y la propiedad z-index), voy cambiando la clase ‘active’ según la tabla que se ha seleccionado para que sea visible. Implemento esta función para cada etiqueta ‘div’ que van a mostrar las diferentes tablas al ser pulsadas.

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

    $("table td").css({
        'border': '2px solid white',
        'text-align': 'left',
        'width': '400px',
        'padding': '20px'
    });


    $("table").css({
        'margin-left': 'auto',
        'margin-right': 'auto',
        'background-color': '#BDB76B',
        'border-collapse': 'collapse',
        'width': '70%'
    });
    $("th").css('background-color', 'gray');
    $('td').hover( function() {
        $(this).parent().children().css("background-color", "#DDD");
        $(this).css("background-color", "#FAFAD2")
    });
    $('tr').mouseleave( function() {
        $(this).children('td').css("background-color", "#BDB76B");
    });
```

```
$('#T1').on('click',function(){
    $('.btActive').css('background-color', 'white');
    $(this).css('background-color', '#DDD');
    var btActive = $('.btActive');
    var btT1 = $(this);
    btActive.removeClass('btActive');
    btT1.addClass('btActive');
    var tableActiva = $('.active');
    var table1 = $('#table1');
    tableActiva.removeClass('active').css('z-index', -10);
    table1.addClass('active').css('z-index', 10);
});

$('#T2').on('click',function(){
    $('.btActive').css('background-color', 'white');
    $(this).css('background-color', '#DDD');
    var btActive = $('.btActive');
    var btT2 = $(this);
    btActive.removeClass('btActive');
    btT2.addClass('btActive');
    var tableActiva = $('.active');
    var table2 = $('#table2');
    tableActiva.removeClass('active').css('z-index', -10);
    table2.addClass('active').css('z-index', 10);
});
```

Apartado 2.1. Personalización condicionada de campos SELECT en formularios.

Primero verifico si el campo input referente a los países coincide con alguno que tenga ciudades guardadas. En tal caso recojo toda la información del fichero de texto correspondiente (habiéndolo abierto mediante tecnología AJAX) y con el método split() separo los saltos de línea para transformar el contenido en un array. Antes de crear las opciones de la lista desplegable, borro todos los elementos de dicha lista por si se hubiese rellenado previamente con otros datos.

```
function cities(input){
    if(input=="Francia")
    {
        var city = new XMLHttpRequest();
        city.onreadystatechange = function(){
            if(this.readyState == 4 && this.status == 200)
            {
                //document.getElementById("cities").innerHTML = this.responseText;
                var txt = this.responseText;
                txt = txt.split('\n');
                //alert(txt);
                var x = document.getElementById("mySelect");
                while(x.firstChild){ //primero vacío
                    x.removeChild(x.firstChild);
                }
                for(var i=0;i<txt.length;i++)
                {
                    if(txt[i]){
                        var option = document.createElement("option");
                        option.text = txt[i];
                        x.add(option);
                    }
                }
            }
        };
        city.open("GET", "cities/Francia.txt", true);
        city.send();
    }
    if(input=="Dinamarca")
    {
        var city = new XMLHttpRequest();
        city.onreadystatechange = function(){
            if(this.readyState == 4 && this.status == 200)
            {
                //document.getElementById("cities").innerHTML = this.responseText;
                var txt = this.responseText;
                txt = txt.split('\n');
                //alert(txt);
                var x = document.getElementById("mySelect");
                while(x.firstChild){
                    x.removeChild(x.firstChild);
                }
                for(var i=0;i<txt.length;i++)
                {
                    if(txt[i]){
                        var option = document.createElement("option");
                        option.text = txt[i];
                        x.add(option);
                    }
                }
            }
        };
        city.open("GET", "cities/Dinamarca.txt", true);
        city.send();
    }
```

Apartado 2.2. Muestra de tablas con Ajax.

Con respecto a la carga mediante el fichero XML a partir del evento *onload* de la etiqueta *body*, abro mediante tecnología AJAX el fichero correspondiente. Recojo en una variable toda la información contenida en el fichero y creo la fila de cabeceras. Recogemos todos los elementos con el tag 'ESTACION' y los recorremos creando una fila para cada uno y una celda para cada componente. Al final, introducimos la tabla creada en otra que está esperando vacía en nuestro documento principal.

```
<script>

    function cargarEstaciones(){
        var xhr = new XMLHttpRequest();
        xhr.onreadystatechange = function () {
            if(this.readyState == 4 && this.status == 200){
                cargarXML(this);
            }
        };
        xhr.open("GET", "xml/tablas.xml", true);
        xhr.send();
    }

    function cargarXML(xml){
        var docXML = xml.responseXML;
        var tabla = "<tr><th>Ciudad</th><th>Temperatura</th><th>Humedad</th><th>Ruido
        //alert('y');
        var ciudad = docXML.getElementsByTagName("ESTACION");
        for(var i = 0; i<ciudad.length; i++)
        {
            tabla += "<tr><td>";
            tabla += ciudad[i].getElementsByTagName("CIUDAD")[0].textContent;
            tabla += "</td><td>";
            tabla += ciudad[i].getElementsByTagName("TEMPERATURA")[0].textContent;
            tabla += "</td><td>";
            tabla += ciudad[i].getElementsByTagName("HUMEDAD")[0].textContent;
            tabla += "</td><td>";
            tabla += ciudad[i].getElementsByTagName("RUIDO")[0].textContent;
            tabla += "</td><td>";
            tabla += ciudad[i].getElementsByTagName("LUZ")[0].textContent;
            tabla += "</td><td>";
            tabla += ciudad[i].getElementsByTagName("ILUMINACION")[0].textContent;
            tabla += "</td><tr>";
        }
        document.getElementById("table2").innerHTML = tabla;
    }
</script>
```

```
<ESTACIONES>
    <ESTACION>
        <CIUDAD>Cuenca</CIUDAD>
        <TEMPERATURA>14</TEMPERATURA>
        <HUMEDAD>30</HUMEDAD>
        <RUIDO>80</RUIDO>
        <LUZ>101</LUZ>
        <ILUMINACION>FF00FF</ILUMINACION>
    </ESTACION>
    <ESTACION>
        <CIUDAD>Gerona</CIUDAD>
        <TEMPERATURA>35</TEMPERATURA>
        <HUMEDAD>82</HUMEDAD>
        <RUIDO>342</RUIDO>
        <LUZ>250</LUZ>
        <ILUMINACION>F0F0F0</ILUMINACION>
    </ESTACION>
    <ESTACION>
        <CIUDAD>Barcelona</CIUDAD>
        <TEMPERATURA>32</TEMPERATURA>
        <HUMEDAD>23</HUMEDAD>
        <RUIDO>120</RUIDO>
        <LUZ>202</LUZ>
        <ILUMINACION>FFF000</ILUMINACION>
    </ESTACION>
    <ESTACION>
        <CIUDAD>Huelva</CIUDAD>
        <TEMPERATURA>13</TEMPERATURA>
        <HUMEDAD>77</HUMEDAD>
        <RUIDO>132</RUIDO>
        <LUZ>132</LUZ>
        <ILUMINACION>00000F</ILUMINACION>
    </ESTACION>
    <ESTACION>
        <CIUDAD>Huesca</CIUDAD>
        <TEMPERATURA>15</TEMPERATURA>
        <HUMEDAD>54</HUMEDAD>
        <RUIDO>99</RUIDO>
        <LUZ>123</LUZ>
        <ILUMINACION>000F0F</ILUMINACION>
    </ESTACION>
</ESTACIONES>
```

En el caso del fichero JSON había que programar en JQuery. Mediante el método getJSON() de la tecnología AJAX, abrimos el fichero y para cada llave (equivalente a recoger cada tag 'ESTACION') formamos una fila y celdas para cada dato que la compone. Mediante el método append() introducimos los datos en la tabla correspondiente que ya cuenta con su fila de cabeceras.

```
$.getJSON("json/tablas.json", function(data){
    var estacion ='';
    $.each(data, function(key, value){
        estacion += '<tr>';
        estacion += '<td>'+value.ciudad+'</td>';
        estacion += '<td>'+value.temperatura+'</td>';
        estacion += '<td>'+value.humedad+'</td>';
        estacion += '<td>'+value.ruido+'</td>';
        estacion += '<td>'+value.luz+'</td>';
        estacion += '<td>'+value.iluminacion+'</td>';
        estacion += '</tr>';
    });
    $('#table3').append(estacion);
});
```

```
[
    {
        "ciudad":"Jaén",
        "temperatura":"35",
        "humedad":"23",
        "ruido":"104",
        "luz":"202",
        "iluminacion":"FF00FF"
    },
    {
        "ciudad":"Lugo",
        "temperatura":"7",
        "humedad":"89",
        "ruido":"186",
        "luz":"90",
        "iluminacion":"FFFFF0"
    },
    {
        "ciudad":"Barcelona",
        "temperatura":"13",
        "humedad":"77",
        "ruido":"289",
        "luz":"178",
        "iluminacion":"F0F0F0"
    },
    {
        "ciudad":"Málaga",
        "temperatura":"36",
        "humedad":"13",
        "ruido":"197",
        "luz":"200",
        "iluminacion":"00000F"
    },
    {
        "ciudad":"Murcia",
        "temperatura":"34",
        "humedad":"22",
        "ruido":"143",
        "luz":"207",
        "iluminacion":"FFF000"
    }
]
```