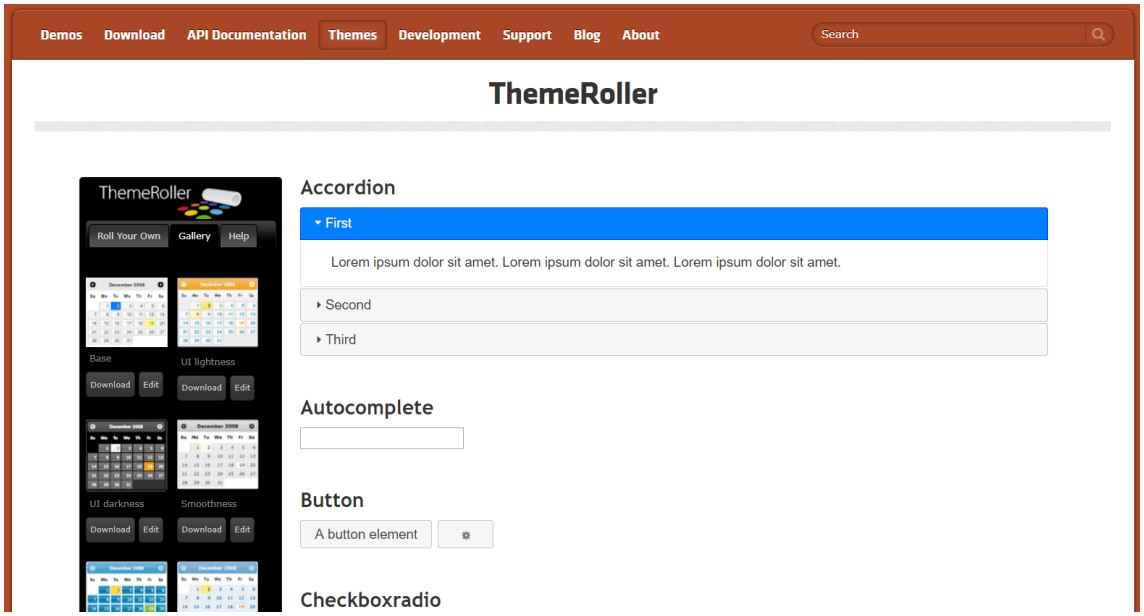
IMPLEMENTANDO LAS TAREAS

Tarea 1: *Lee el artículo Using Multiple jQuery UI Themes on a Single Page y pruébalo. Utilízalo con el widget Dialog usando un rango variado de propiedades y métodos.*

Como se ha implementado:

Para poder usar varios temas de jQuery UI en una misma página deberemos primero irnos a la página de temas.



Una vez elijamos un par que nos gusten, le damos a Download y en el apartado de CSS Scope deberemos poner el tipo de etiqueta que tendrá el tema o un nombre de clase.

En este caso he puesto div.blitzer. de este modo el tema se aplicara a todos los div que tengan una clase llamada blitzer.

Este paso deberemos hacerlo tantas veces como temas queramos y con el nombre de la clase diferente, por comodidad le ponemos el nombre del tema.

temaBlitzer	01/03/2018 20:03	File folder
temaFrog	01/03/2018 20:03	File folder
temaLuv	04/03/2018 12:18	File folder

Es nuestra carpeta del proyecto iremos poniendo las de los temas que hemos descargado.

Nos iremos a nuevo archivo index.html y pondremos los css de los temas que queramos usar como se puede observar en la imagen de abajo.

```
<link rel="stylesheet" href="temaFrog/jquery-ui.css" type="text/css">
<link rel="stylesheet" href="temaBlitzer/jquery-ui.css" type="text/css">
<link rel="stylesheet" href="temaLuv/jquery-ui.css" type="text/css">
```

Ahora ya solo es cuestión de elegir los elementos de jquery ui que deseemos poner con un tema diferente al de por defecto y ponerles una clase de un tema.

En este caso ponemos luv.

```
<div class="form-group col-md-6 luv sexoTamaño">
    <label>Sexo</label>
    <label for="radio">Hombre</label>
    <input class="input" type="radio" name="radio" id="radio">
    <label for="checkbox">Mujer</label>
    <input class="input" type="radio" name="checkbox" id="checkbox">
</div>
```

Como resultado vemos como en un dialog con un tema blitzer tiene en su interior un checkradio con un tema luv.

Tarea 2: *Utiliza el widget Autocomplete con una fuente remota de datos en formato JSON.*

Puedes basar tu código en los ejemplos de "demo JqueryUI", pero debe ser una API diferente.

Como se ha implementado:

Esta tarea es la más compleja ya que deberemos tener un autocomplete el cual tenga un array en un json.

Primero deberemos crear dentro de la caja de script de nuestro index.html, una petición Ajax a un archivo json el cual contiene las provincias de España.

Estas provincias se guardarán en un array.

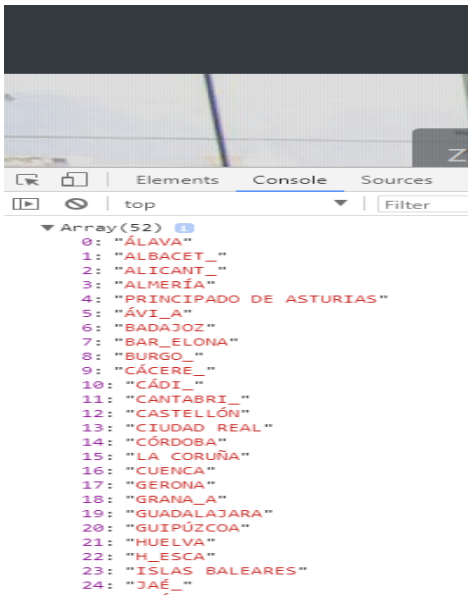
```
var array = []

$.ajax({
    url: 'https://cors-anywhere.herokuapp.com/https://apivl.geoapi.es/provincias?type=JSON&key=&sandbox=1%27',
    type: 'GET',
    crossDomain: true,
    success: function(resolve) {
        resolve.data.map(function(item) {
            array.push(item.PRO);
        })
    },
    error: function() {
        alert('Failed!');
    }
});

console.log(array);
```

Podemos observar dándole a f12 en nuestra pagina y en la sección de consola como existe un array con todas las provincias.

De ahí iremos sacando el autocomplete.



También será importante indicarle a partir de cuantas letras nos empezara a mostrar los posibles resultados además de indicarle que las palabras del array tienen que empezar por lo escrito por nosotros ya que si no, nos mostrara también los que contengan las palabras que hemos escrito.

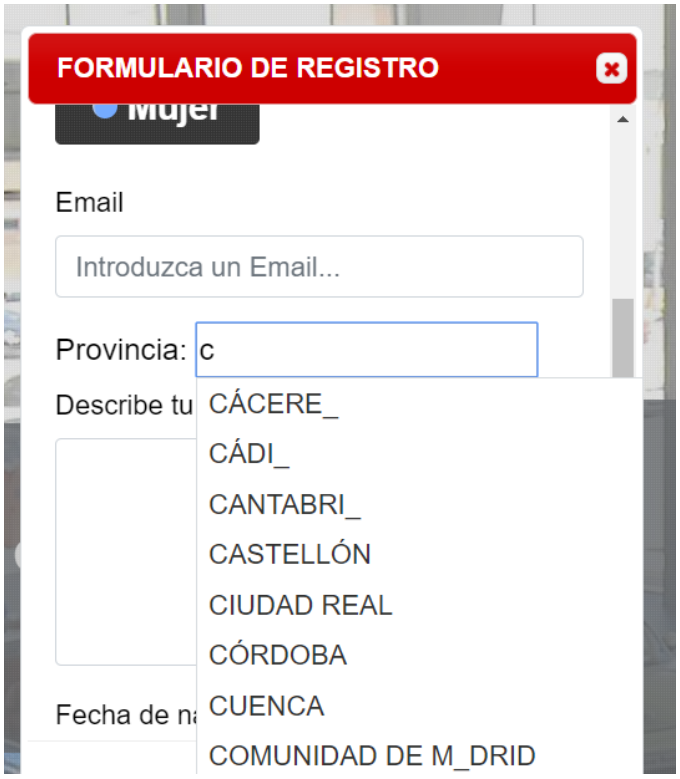
```
$(function() {

    $("#tags").autocomplete({
        minLength: 1,
        source: function(request, response) {
            var matcher = new RegExp("^" + $.ui.autocomplete.escapeRegex(request.term), "i");
            response($.grep(array, function(item) {
                return matcher.test(item);
            }));
        },
        select: function(event, ui) {
            $("input.autocomplete").val(ui.item);
        }
    });
});
```

En la parte de html simplemente tendremos que tener una clase ui-widget con un for e id tags para poder llamar a nuestra función.

```
<div class="ui-widget frog">
    <label for="tags">Provincia: </label>
    <input id="tags">
</div>
```

Aquí podemos ver el resultado:



Tarea 3: *Usando el Widget Datapicker, implementar algo similar al selector de fechas de la interfaz de Booking.com.*

Como se ha implementado:

Queremos tener un calendario similar al de booking, es decir, que muestre los meses de 2 en 2, que este en español y demás características.

Tendremos una funcionar en la que indicaremos como deseamos que sea nuestro calendario.

Por ejemplo, poniendo numberOfMonths:2 conseguimos que nos aparezcan los meses de 2 en 2.

Tambien con la opción de buttonImage podemos poner en vez del botón con los 3 puntos una imagen.

Hemos incorporado una animación con showAnim.

```
$(function() {

    $( "#fecha" ).datepicker({
        prevText: "Earlier",
        autoSize: true,
        changeYear: true,
        showOn: "both",
        numberOfMonths: 2,
        buttonImage: "http://images.revolutionreservations.com/calendar.gif",
        showButtonPanel: true,
        showAnim: 'slideDown'
    });
```

Por defecto el calendario esta en ingles por lo que nos debemos de encargar de ponerlo en español. La forma de hacerlo es la siguiente:

Configuramos por defecto como seria todo en español y una vez lo hacemos indicamos que nuestro calendario por defecto tendrá de idioma el español, asi conseguimos que las palabras las saque de nuestra configuración anterior la cual hicimos.

```
$.datepicker.regional.es = {
closeText: "Cerrar",
prevText: "&#x3C;Ant",
nextText: "Sig&#x3E;",
currentText: "Hoy",
monthNames: [ "enero","febrero","marzo","abril","mayo","junio",
"julio","agosto","septiembre","octubre","noviembre","diciembre" ],
monthNamesShort: [ "ene","feb","mar","abr","may","jun",
"jul","ago","sep","oct","nov","dic" ],
dayNames: [ "domingo","lunes","martes","miércoles","jueves","viernes","sábado" ],
dayNamesShort: [ "dom","lun","mar","mié","jue","vie","sáb" ],
dayNamesMin: [ "D","L","M","X","J","V","S" ],
weekHeader: "Sm",
dateFormat: "dd/mm/yy",
firstDay: 1,
isRTL: false,
showMonthAfterYear: false,
yearSuffix: "" };
$.datepicker.setDefaults($.datepicker.regional['es']);
```

Observamos el resultado:

Tarea 4: *Selecciona al menos dos efectos diferentes de JqueryUI e impleméntalos usando todas las opciones disponibles para cada uno de ellos.*

Como se ha implementado:

En nuestro trabajo hemos utilizado 3 efectos, 1 para nuestro calendario y otros dos para cuando abramos y cerremos el dialog.

Con show y effect le decimos al dialog que tenga cierta animación cuando lo mostremos.

Y con hide y effect sería lo mismo, pero cuando lo cerremos.

Podemos ponerle una duración si lo deseamos para configurarlo mas.

```
$( "#dialog" ).dialog({
    modal:true,
    width:"400px",
    height:"200px",
    buttons:{
        "Cerrar":aceptar
    },
    autoOpen: false,
    show: {
        effect: "blind",
        duration: 1000
    },
    hide: {
        effect: "explode",
        duration: 1000
    }

    });
    $( "#opener" ).on( "click", function() {
    $( "#dialog" ).dialog( "open" );
    });
    $( ".input" ).checkboxradio();
```

Para nuestro calendario le añadimos en la función una opción de showAnim y el tipo de efecto que deseemos. En nuestro caso al abirlo se desplegará de arriba abajo y cuando lo cerremos se plegara de abajo arriba.

```
$( "#fecha" ).datepicker({
    prevText: "Earlier",
    autoSize: true,
    changeYear: true,
    showOn: "both",
    numberOfMonths: 2,
    buttonImage: "http://im
    showButtonPanel: true,
    showAnim: 'slideDown'
```