

## LISTAS DESPLEGABLES ENCADENADAS

1. Crea un script que cargue de forma dinámica, mediante AJAX, la lista de provincias de un país y la lista de los municipios de cada provincia seleccionada.

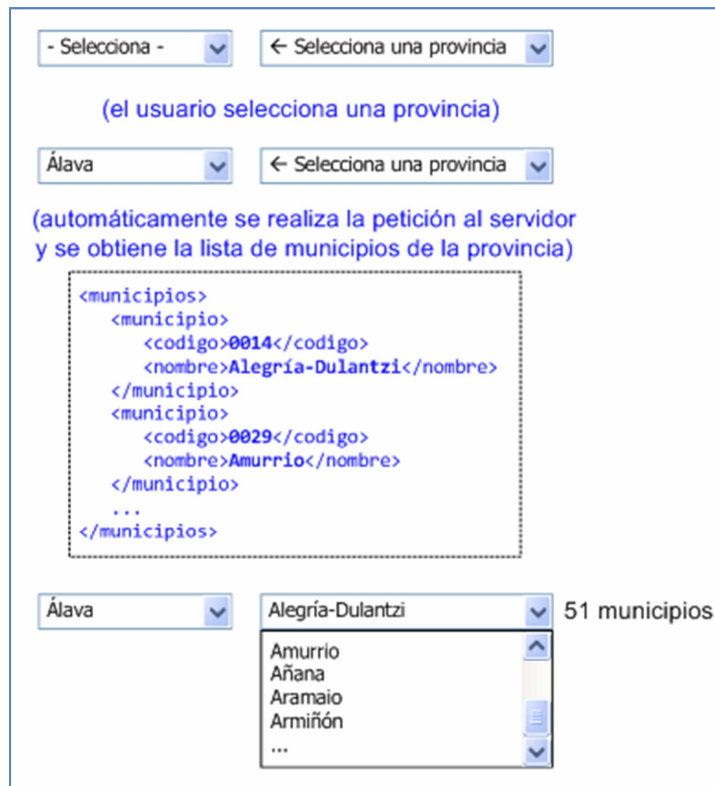
- 1) Definir el código HTML de las dos listas desplegables vacías.
- 2) Cuando se cargue la página, cargar la lista de provincias en la primera lista desplegable: El script del servidor se llama **cargaProvinciasXML.php**. El formato de la respuesta es XML, con la siguiente estructura:

```
<provincias>
  <provincia>
    <codigo>01</codigo>
    <nombre>Álava</nombre>
  </provincia>
  . . .
</provincias>
```

Para insertar las opciones en la lista desplegable, se pueden utilizar dos técnicas:

1. Propiedad `innerHTML` de la lista y código HTML de cada etiqueta `<option>`.
  2. Crear elementos de tipo opción (`new Option(nombre, valor)`) y añadirlo al array `options[]` de la lista desplegable.
- 3) Añadir de forma semántica el evento adecuado a la lista de provincias para que cuando se seleccione una provincia, se carguen automáticamente todos sus municipios correspondientes en la otra lista.
  - 4) Cuando se seleccione una determinada provincia, se carga mediante AJAX la lista completa de municipios en la otra lista desplegable. El script del servidor se llama **cargaMunicipiosXML.php**. El parámetro que se debe enviar al servidor es el código de la provincia y el parámetro se llama provincia. El método que espera el servidor es POST. El formato de la respuesta es XML, con la siguiente estructura:

```
<municipios>
<municipio>
<codigo>0014</codigo>
<nombre>Alegría-Dulantzi</nombre>
</municipio>
...
</municipios>
```



- Selecciona -    ← Selecciona una provincia

(el usuario selecciona una provincia)

Álava    ← Selecciona una provincia

(automáticamente se realiza la petición al servidor  
y se obtiene la lista de municipios de la provincia)

```
<municipios>
  <municipio>
    <codigo>0014</codigo>
    <nombre>Alegria-Dulantzi</nombre>
  </municipio>
  <municipio>
    <codigo>0029</codigo>
    <nombre>Amurrio</nombre>
  </municipio>
  ...
</municipios>
```

Álava    Alegria-Dulantzi    51 municipios

Amurrio  
Añana  
Aramaio  
Armiñón  
...

2. Modificar el ejercicio anterior para soportar las respuestas del servidor en formato JSON. Los cambios introducidos son los siguiente:

1) El script del servidor utilizado para cargar las provincias se llama **cargaProvinciasJSON.php** y la respuesta del servidor tiene el siguiente formato:

```
{ "01": "Álava", "02": "Albacete", "03": "Alicante", ... }
```

2) El script del servidor utilizado para cargar los municipios se llama **cargaMunicipiosJSON.php** y la respuesta del servidor tiene el siguiente formato:

```
{ "0014": "Alegria-Dulantzi", "0029": "Amurrio", ... }
```

## AUTOCOMPLETAR

---

3. A partir del formulario proporcionado, añadir la opción de autocompletar el nombre del municipio que está escribiendo el usuario. El esquema del funcionamiento propuesto es el siguiente:

- 1) Al cargar la página, se debe crear un elemento HTML de tipo `<div>` en el que se van a mostrar las sugerencias enviadas por el servidor.

Además, se debe establecer el evento de teclado adecuado en el cuadro de texto y también se debe posicionar el cursor en ese cuadro de texto para poder escribir en el directamente.

- 2) Cuando se pulse una tecla sobre el cuadro de texto, se debe ejecutar la función `autocompleta()`. Desde esta función, se debe llamar a la función responsable de obtener la lista de municipios del servidor. El script se llama [autocompletaMunicipios.php](#), el parámetro que se envía mediante POST, se llama `municipio` y debe contener la cadena de texto escrita por el usuario. El servidor responde con un array en formato JSON con la lista de municipios cuyo nombre comienza por el texto enviado. Ejemplo de respuesta del servidor:

```
[ "Alegoría", "Amurrio", "Añana", "Aramaio", "Armiñón", ... ]
```

- 3) Una vez obtenido el array de sugerencias, se debe mostrar en forma de lista de elementos (etiqueta `<ul>` de HTML). Para transformar el array en la lista `<ul>`, modificar el prototype del objeto `Array` y añadir una función específica que realice la transformación.
- 4) Modificar la función `autocompleta()` para tener en cuenta 3 teclas especiales: las flechas superior e inferior y la tecla Enter. Cuando se utilizan las flechas del teclado hacia arriba y hacia abajo, se van seleccionando los elementos de la lista (con el cursor sobre el cuadro de texto). Cuando se pulsa el Enter, se selecciona el elemento copiando su valor al cuadro de texto y ocultando la lista de sugerencias. (Pista: variable global `elementoSeleccionado`)
- 5) Para mejorar el rendimiento de la aplicación, añadir una caché para las sugerencias. Cada vez que se recibe una lista de sugerencias del servidor, se almacena en un objeto que relaciona el texto que ha introducido el usuario y la respuesta del servidor. Ejemplo:

```
{
```

```
"a": ["Ababuj", "Abades", "Abadía", "Abadín", "Abadiño",
"Abáigar", "Abajas", "Abaltzisketa", "Abánades", "Abanilla",
"Abanto y Ciérvana-Abanto Zierbena", "Abanto", "Abarán",
"Abarca de Campos", "Abárzuza", "Abaurregaina/Abaurrea
Alta", "Abaurrepea/ Abaurrea Baja", "Abegondo", "Abejar",
"Abejuela", "Abella de la Conca"],

"al": ["Alacón", "Aladrén", "Alaejos", "Alagón", "Alaior",
"Alájar", "Alajeró", "Alameda de Gardón (La)", "Alameda de
la Sagra", "Alameda del Valle", "Alameda", "Alamedilla
(La)", "Alamedilla", "Alamillo", "Alaminos", "Alamús (Els)",
"Alange", "Alanís", "Alaquàs", "Alar del Rey", "Alaraz"],
...
}
```

De esta forma, antes de realizar una petición al servidor, se comprueba si ya está disponible una lista de sugerencias para ese texto. Además, cuando se realiza una consulta al servidor, la respuesta se almacena en la cache para su posible reutilización futura.

- 6) Mostrar un mensaje adecuado cuando el servidor devuelva un array vacío por no haber sugerencias para el texto introducido por el usuario.

