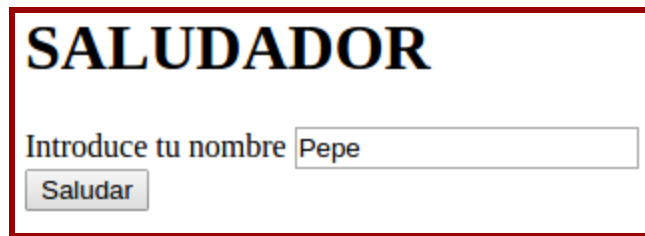


# Ejercicios de acceso eficiente al servidor (AJAX)

1. Generar un script que muestre una salida con el siguiente aspecto:

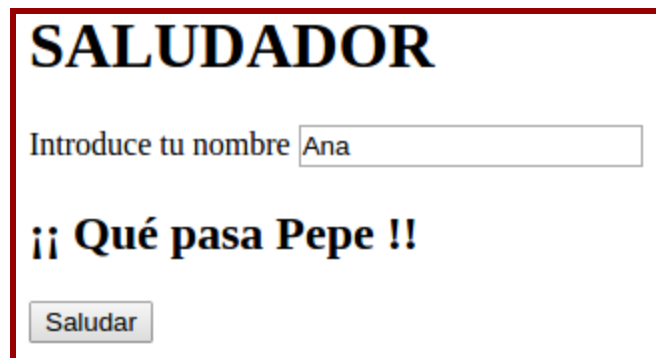


**SALUDADOR**

Introduce tu nombre

Al pulsar el botón se invocará (vía AJAX) un script que construirá aleatoriamente un saludo de la forma “Hola <nombre>”, “Qué tal estás <nombre>” ó “¡¡Qué pasa <nombre>!!”, entre etiquetas <h1>, <h2> ó <h3> también escogidas aleatoriamente. Dicho saludo se insertará entre el cuadro de texto y el botón.

Se podrán seguir introduciendo nombres y saludando hasta que se quiera.



**SALUDADOR**

Introduce tu nombre

**¡¡ Qué pasa Pepe !!**

2. Generar dinámicamente dos controles de selección que muestren: uno las comunidades autónomas de España, y el otro las provincias de la comunidad seleccionada. La recuperación de las provincias al cambiar la comunidad se realizará vía Ajax.



**Comunidades autónomas**

ccaa

provincias

**Escoge una comunidad autónoma**

Observa el cambio de provincias vía AJAX

Inicialmente el script mostrará la primera comunidad (por orden alfabético) y la primera provincia. A partir de ahí, se detectará el evento de cambio de comunidad en el select (evento “onChange”) para lanzar la consulta vía AJAX que devolverá el código HTML del select de provincias para esa comunidad en concreto. Dicho código sustituirá al que hubiera anteriormente.

La información de nombres de comunidades y de provincias se puede implementar en un script aparte, mediante un array asociativo (como el adjunto a continuación). En un caso real estaría en una Base de Datos.

```
[ 'Andalucía' => [ 'Almería', 'Cádiz', 'Córdoba', 'Granada', 'Huelva', 'Jaén', 'Málaga', 'Sevilla' ], 'Aragón' => [ 'Huesca', 'Teruel', 'Zaragoza' ], 'Asturias (Principado de)' => [ 'Asturias' ], 'Canarias' => [ 'Palmas (Las)', 'Santa Cruz de Tenerife' ], 'Cantabria' => [ 'Cantabria' ], 'Castilla y León' => [ 'Ávila', 'Burgos', 'León', 'Palencia', 'Salamanca', 'Segovia', 'Soria', 'Valladolid', 'Zamora' ], 'Castilla-La Mancha' => [ 'Albacete', 'Ciudad Real', 'Cuenca', 'Guadalajara', 'Toledo' ], 'Cataluña' => [ 'Barcelona', 'Gerona', 'Lérida', 'Tarragona' ], 'Ceuta (Ciudad de)' => [ 'Ceuta' ], 'Comunidad Valenciana' => [ 'Alicante', 'Castellón', 'Valencia' ], 'Extremadura' => [ 'Badajoz', 'Cáceres' ], 'Galicia' => [ 'Coruña (La)', 'Lugo', 'Orense', 'Pontevedra' ], 'Islas Baleares' => [ 'Islas Baleares' ], 'Madrid (Comunidad de)' => [ 'Madrid' ], 'Melilla (Ciudad de)' => [ 'Melilla' ], 'Murcia (Región de)' => [ 'Murcia' ], 'Navarra (Comunidad Foral de)' => [ 'Navarra' ], 'País Vasco' => [ 'Álava', 'Guipúzcoa', 'Vizcaya' ], 'Rioja (La)' => [ 'Rioja (La)'] ]
```

### 3. Realizar una aplicación con el siguiente aspecto

La aplicación simula un formulario de traducciones de palabras, es decir, no debe hacer nada en especial cuando se pulse el botón de submit. Sin embargo al cambiar el radio button de las banderas, se cambiarán las etiquetas de los input-text y del botón por sus correspondientes en español, francés o inglés.

Hacer tres versiones distintas del script,

- una en la que se tenga que redibujar entero para cambiar el texto de las etiquetas.
- Otra en la que se invoque vía AJAX el conjunto de etiquetas para ese idioma,
- y otra en la que en la primera carga se provean todas las posibles traducciones y el cambio de etiquetas se haga únicamente vía Javascript (y no AJAX)

PISTA1: Para saber en Javascript el value de un radio seleccionado

`document.querySelector('input[name="<nombreRadio>"]:checked').value;`

PISTA2: En la versión AJAX, para evitar XML, podemos “empaquetar” los datos con `implode('separador',array)` en PHP, y desempaquetarlos con `texto.split('separador'):array` de javascript.

Para las traducciones se puede utilizar el siguiente array asociativo:

```
$BDetiquetas=[ "ES"=>[ "Palabra", "Traducción", "Enviar" ], "EN"=>[ "Word", "Translation", "Send" ], "FR"=>[ "Mot", "Traduction", "Envoyer" ] ]
```

### 4. Hacer una aplicación con el siguiente aspecto.




Usuario   
 Contraseña




Introduce a number (1-10)   
  
 My number: 3  
 Your number: 5  
 Sorry, your number is higher

Una primera pantalla de login (con usuario / password) en la que el usuario podrá introducir sus credenciales. Si éstas son correctas (p.ej validándolas contra un array asociativo) aparecerá una segunda pantalla en la que se podrá jugar a un juego simple en el que hay que adivinar un número del 1 al 10. Al pulsar “Jugar”, se calculará un número al azar e informará de si te has pasado, quedado corto o acertado. Se podrá jugar todas las veces que se quiera.

Al pulsar el botón “Salir” se volverá a la pantalla de login y en el nombre de usuario aparecerá el último usuario que jugó.

Las etiquetas de la aplicación se podrán conmutar entre español e inglés en cualquiera de las dos pantallas y el cambio de etiquetas se hará vía AJAX o en PHP según se considere necesario.

Si el usuario o la contraseña son incorrectos, se mostrará una pantalla informativa (en el mismo idioma que estuviera la pantalla de login) con un botón (cuya etiqueta estará también en el idioma adecuado) que permitirá volver a la pantalla de login para volver a intentar el proceso de autenticación

## Usuario o contraseña no válidos

5. Realizar un programa con el siguiente aspecto

## Lista de empleados

Nombre ▼

Nombre	Apellido	Teléfono
Pepe	García	637 654 654
Ana	García	637 111 654
Manolo	González	637 654 654

Mostrará una lista de empleados que obtendrá de una ficticia Base de Datos (almacenada en un array asociativo en el lado servidor).

La aplicación permitirá introducir una cadena de texto y escoger entre “Nombre”, “Apellido” y “Teléfono” en un select.

Al pulsar el botón “Filtrar por” se generará una petición AJAX que enviará los registros de la BD que cumplan el filtro, y los sustituirá por el listado anterior.

6.