

EJERCICIOS PRÁCTICOS TEMA 7

1. Crea un formulario que permita escribir un nombre, apellidos, email y un NIF, con una estética similar a la de la siguiente imagen:

Registrar usuario

Nombre

Apellidos

Email

NIF

Enviar

Inicialmente los cuadros de entrada de texto poseerán un borde rojo para indicar que el contenido aún no es válido. El nombre y apellidos se consideran válidos si al menos hay dos letras. El email si hay un símbolo de @ y a su izquierda y derecha al menos una letra. Se hacen las validaciones a medida que vamos escribiendo, con cada tecla pulsada. La validación del NIF utiliza un servicio alojado en un servidor. El botón de comprobar permanece desactivado hasta que todos los cuadros de texto sean válidos. Cuando lo sean el botón se activa. Al hacer clic en el botón comprobar, simplemente aparecerá un mensaje indicando que todo es correcto. Al hacer clic en el mensaje comenzamos de nuevo.

2. La página <https://randomuser.me/> Permite obtener datos aleatorios de personas pensando en que los desarrolladores y otros profesionales puedan utilizarlos en sus pruebas y test. Las instrucciones de la API de este servicio gratuito están en la url: <https://randomuser.me/documentation> . En todo caso la idea es hacer peticiones vía GET a la url: <https://randomuser.me/api/> . Se pueden pasar para indicar cuántos usuarios aleatorios deseamos, el sexo, política de contraseñas, páginas, formato de respuesta, etc. En la página de documentación viene un ejemplo de la estructura JSON de las respuestas. Como resumen indicamos que es un objeto formado por dos propiedades: results e info. La primera es un array donde cada elemento lo forma un objeto con los datos del usuario aleatorio. La propiedad info contiene otros detalles entre los que destaca una semilla que permite repetir una petición con los mismos datos y datos de paginación. La aplicación mostrará la foto, nombre, apellido, email, dirección y estado al que pertenece el usuario. Cada vez que actualicemos la página se pedirá otro usuario.
3. Usaremos el mismo servicio de usuarios aleatorio para esta práctica. Si pasamos un parámetro llamado results con valor 10, conseguiremos obtener datos de 10 usuarios aleatorios. Mostraremos esos datos en dos filas. Distinguiremos el fondo de los pares e impares y, además, en cada usuario colocaremos un botón con el texto “Cambiar”. Cuando hagamos clic en el botón cambiar, el usuario en el que está situado el botón, se cambiará. Mostraremos el mensaje “Cargando...” al principio, mientras llegan los datos de los 10 usuarios. También cuando pulsemos cambiar en el usuario a modificar, escribiremos “Esperando usuario nuevo...” en la celda de ese usuario (habremos, antes, quitado los datos del usuario) hasta que lleguen los nuevos datos.

Desarrollo Web en Entorno Cliente

4. La agencia meteorológica española AEMET dispone de una API para datos abiertos en la que proporciona numerosas posibilidades de obtener información meteorológica. Para poder obtener los datos de esta agencia necesitamos darnos de alta como desarrolladores en la dirección: <https://opendata.aemet.es/centrodedescargas/altaUsuario> . Las instrucciones de funcionamiento de la API de AEMET están en: <https://opendata.aemet.es/dist/index.html> . En el alta necesitamos indicar un email que tendremos que confirmar para que se nos conceda una API key, una clave única que se otorga a cada desarrollador. esa API la debemos enviar cada vez que queramos hacer peticiones a la AEMET. Nuestra aplicación, inicialmente, muestra un cuadro de texto en el que introducir nuestra API y un botón de cargar mapa. Tras pegar nuestra API y pulsar el botón “Cargar mapa”, solicitaremos el mapa meteorológico del día en la dirección: <https://opendata.aemet.es/opendata/api/mapasygraficos/analisis> . Se quitarán los controles de formulario y se mostrará en horizontal el mapa. Hay que recordar debemos enviar vía GET un parámetro llamado apikey, con nuestra clave, la cual se obtiene del formulario. Los datos que llegan como nos ofrecen un objeto JSON donde la propiedad llamada datos, es una URL a la imagen del mapa. La petición http nos proporciona ese enlace, pero no el mapa en sí. Por ello, hay una segunda petición a esa URL devuelta por la propiedad datos que es la que nos proporciona el mapa.