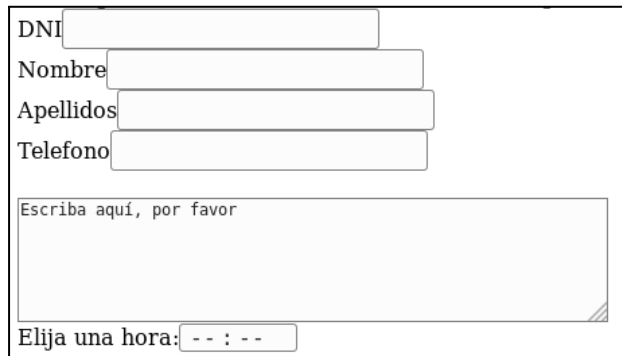


## EJERCICIOS FORMULARIOS Y ALMACENAMIENTO EN JAVASCRIPT

### EJERCICIO Form1

Crea el siguiente formulario:



Formulario de registro Form1:

- DNI:
- Nombre:
- Apellidos:
- Teléfono:
- Comentario: 

Escriba aquí, por favor
- Elija una hora:

Valida el contenido introducido en todos campos según las siguientes especificaciones:

- DNI (99.999.999-X)
- Nombre (Mínimo 1 nombre, Máximo 2 nombres)
- Apellidos (Mínimo 1 apellido, Máximo 2 apellidos)
- Teléfono (+34 954556817)
- Comentario (texto obligatorio de tamaño máximo 250 caracteres)
- Hora (hh:mm)

Validar también que no puedan dejarse en blanco.

NOTA: puedes utilizar tanto validaciones HTML como lo aprendido en Javascript.

En caso de que se informe correctamente guardar la información en un objeto y obtener el JSON correspondiente, en caso contrario mostrar un mensaje de error.

### EJERCICIO Form2

Crea el siguiente formulario:



Formulario de contacto Form2:

- Escribe tu nombre:
- Escribe tus apellidos:
- Escribe tu Email:
- Web Personal:
-

Valida el contenido introducido en todos campos según las siguientes especificaciones:

- Nombre (Mínimo 1 nombre, Máximo 2 nombres)
- Apellidos (Mínimo 1 apellido, Máximo 2 apellidos)
- Email (xxxxxx@yyyyy.zzz)
- WEB (http://www.xxxx.yyy)

Validar también que no puedan dejarse en blanco.

NOTA: puedes utilizar tanto validaciones HTML como lo aprendido en Javascript.

En caso de que se informe correctamente guardar la información en un objeto y obtener el JSON correspondiente, en caso contrario mostrar un mensaje de error.

### **EJERCICIO Form3**

Valida el formulario con los siguientes campos y en caso de que sean correctos los datos introducidos se almacenen en un objeto y en caso contrario muestre un mensaje de error.

- DNI (99.999.999-X)
- Nombre (Mínimo 1 nombre y 1 apellido, Máximo 2 nombre y 2 apellidos)
- Fecha de Nacimiento (dd/mm/yyyy)
- Email (xxxxxx@yyyyy.zzz)
- WEB (http://www.xxxx.yyy)
- Contraseña (Entre 8 y 10 caracteres)

Ningún campo podrá dejarse en blanco.

NOTA: puedes utilizar tanto validaciones HTML como lo aprendido en Javascript.

En caso de que se informe correctamente guardar la información en un objeto y obtener el JSON correspondiente, en caso contrario mostrar un mensaje de error.

### **EJERCICIO Form4**

Una empresa de comidas tiene que controlar las bolsas de producto azul que cocinan para distribuir en Albuquerque. Para ello tienen que crear un formulario que almacene la información de cada bolsa y validarlo teniendo en cuenta lo siguiente:

- Fecha de creación: obligatorio y con formato dd/mm/aaaa.
- Cocinero: será un nombre en clave formado por dos letras en mayúscula, un símbolo y cuatro dígitos (ej. WW\$1234)
- Destinatario: estará formado por dos o tres letras mayúsculas correspondientes al estado, un guión bajo, el nombre de la ciudad en minúsculas, dos puntos, y el código de distrito de 4 dígitos (ej. NM\_albuquerque:1234).
- Gramos: será un número del 100 al 5000.

- Composición: estará formado por una cantidad en gramos seguida de dos conjuntos de una o dos letras seguidas o no de un número. (ej. 200gC3OH7)
- Número de cuenta bancaria. Supongamos que un número de cuenta estadounidense tiene el siguiente formato:
  - Dos letras: el valor de cada letra es del 1 al 26 (no hay ñ ni ll).
  - Dos dígitos: debe corresponderse con la suma de la primera letra y la segunda: en caso de que sea menor que 10 se pone el 0 delante.
  - Un guión.
  - Doce dígitos de cuenta.
  - Un guión.
  - Dos dígitos de control: el primero debe ser la suma de los 6 primeros dígitos de la cuenta dividido entre 6 y extrayendo solamente su parte entera; y el último exactamente igual, pero con los 6 siguientes.

Si el número está correcto se colocará en un campo de texto al lado del anterior, pero sin guiones: solamente los números y las letras.

Ningún campo podrá dejarse en blanco.

NOTA: puedes utilizar tanto validaciones HTML como lo aprendido en Javascript.

En caso de que se informe correctamente guardar la información en un objeto y obtener el JSON correspondiente, en caso contrario mostrar un mensaje de error.

## EJERCICIO Cookie1

Crea un fichero que se llame “cookies. y que permita trabajar con las cookies de manera que puedas crear, consultar, actualizar y borrar una cookie, operaciones CRUD.

NOTA: Consulta los métodos tratados en clase y recuerda que el método borrar no ha sido creado aún así que deberás crearlo tú mismo.

## EJERCICIO Cookie2

Modifica el formulario del ejercicio Form1 anterior de la siguiente manera:

- Cada vez que el usuario trate de enviar el formulario y haya algún error, una variable contador (almacenada en una cookie) se incrementará.
- El resultado del número de intentos se reflejará en un campo de texto que se encontrará al final del formulario.
- Si el usuario sale del programa y vuelve a entrar, el campo de texto mostrará el número almacenado en la cookie.
- Junto al campo de texto habrá un botón que, al pulsarlo, permitirá reiniciar el valor de la cookie a 0.

### **EJERCICIO Cookie3**

Modifica el formulario del ejercicio Form2 anterior de forma que cuando se rellene el formulario correctamente y se guarden los datos en un objeto y obtengan su json, también guarde en una cookie los siguientes datos de registro, durante 1 día:

- usuario = nombre + apellido
- email = email

### **EJERCICIO API WEB STORAGE**

Modifica los formularios de los ejercicios anteriores para que se guarde la información registrada usando la API Web Storage de la siguiente forma:

- Ejercicio Form3 en el almacenamiento de sesión del navegador.
- Ejercicio Form4 en el almacenamiento local del navegador.

Añadir un botón “Recargar” que recargue los campos del formulario con la información almacenada.

NOTA: Prueba posteriormente la persistencia de los datos en cada caso.