

## Práctica 4



Adrián Garrido Pantaleón.

31/01/2022

## ÍNDICE

<b>Parte 1: validación de formularios con JS y HTML5 .....</b>	<b>3</b>
▫ Especificaciones generales:.....	3
▫ Validación de DNI y campos anidados:.....	3
▫ Envío automático del formulario.....	4
<b>Parte 2: contenido restringido y animación .....</b>	<b>5</b>
▫ Categoría de noticias oculta.....	5
▫ Animación con capa en movimiento.....	6
<b>Parte 3: etiquetas semánticas y RWD en HTML5 .....</b>	<b>7</b>
▫ Etiquetas semánticas y validación del W3C.....	7
▫ Uso de la etiqueta Picture y RWD.....	7
<b>Valoración personal .....</b>	<b>8</b>

## Parte 1: validación de formularios con JS y HTML5

### ▫ Especificaciones generales:

En este apartado hemos comprobado que el uso de HTML 5 ayuda bastante a la hora de simplificar y sobre todo para la corrección de errores y validaciones.

El uso de JavaScript es más que necesario, pero con la herramienta HTML 5 podemos hacer muchas cosas. Un ejemplo claro es para validar el teléfono se puede hacer mediante JS o de esta manera:

```
<input type="text" class="formulario__input" name="telefono" pattern="^[0-9]{9}$" id="telefono"
placeholder="666300123">
```

Donde se pueden ver etiquetas como *pattern*, que se utiliza para saber como quieres el formato, o *placeholder* para ayudar al usuario a saber en todo momento que tiene que hacer. En este caso no hemos utilizado *required* porque no necesitamos que sea un campo obligatorio.

### ▫ Validación de DNI y campos anidados:

Para la validación del DNI, hemos utilizado una función que comprueba que la letra del DNI corresponde a su número, de no ser así lanzaría un mensaje de error.

```
function validarDni(e) {

    var letras;
    var dni = e.value;
    var numero = dni.substr(0, dni.length - 1);

    var letra = dni.substr(dni.length - 1, 1);

    letra = letra.toUpperCase();
    numero = numero % 23;
    letras = 'TRWAGMYFPDXBNJZSQVHLCKET';
    letras = letras.substring(numero, numero + 1);

    if (letras != letra) {
        document.getElementById("dnimal").innerHTML = "El DNI debe ser correcto.";
        return false;
    } else {
        document.getElementById("dnimal").innerHTML = "";
        return true;
    }
}
```

Las comunidades y provincias se han hecho con *selects* para dar multitud de opciones, y posteriormente con JS hacer que eligiera entre todas las posibilidades.

De esta manera si el usuario escoge la Comunidad Valenciana, se le concede la opción de elegir entre Castellón, Valencia y Alicante.

## ▫ Envío automático del formulario

Para el envío automático del formulario hemos tenido que explorar las opciones de un API externa como EmailJS para que cuando nosotros pulsemos el botón de enviar se nos mande un email con todos los datos del formulario.

Si los datos introducidos no son los correctos, el correo no se enviará y se necesitará corregir los errores.

```
const btn = document.getElementById('button');
emailjs.init('user_Fbs242eLhEur11bKpK9c5');
document.forms['formulario'].onsubmit = function (e) {
  e.preventDefault();
  const objeto = {
    usuario: this['usuario'].value,
    nombre: this['nombre'].value,
    password: this['password'].value,
    correo: this['correo'].value,
    fecha: this['fecha'].value,
    url: this['url'].value,
    telefono: this['telefono'].value,
    comunidades: this['comunidadAutonoma'].value,
    provincia: this['provincia'].value,
    producto: this['Productos'].value,
    prodInt: this['ProductosInteres'].value,
    apellidos: this['apellidos'].value,
    dni: this['dni'].value
  }
  console.log(this["comu"])
  emailjs.send('service_z4ngtsn', 'template_ielos0w', objeto)
    .then(function (response) {
      console.log('SUCCESS!', response.status, response.text);
    }, function (error) {
      console.log('FAILED...', error);
    });
}
```

Rellenamos todos los campos requeridos y le damos al botón de enviar.

<b>Nombre</b> <input type="text" value="Adrián"/>	<b>Apellidos</b> <input type="text" value="Garrido Pantaleón"/>
<b>Usuario</b> <input type="text" value="adri07"/>	<b>DNI</b> <input type="text" value="50368951D"/>
<b>Contraseña</b> <input type="password" value="****"/>	<b>Repetir Contraseña</b> <input type="password" value="****"/>
<b>Correo Electrónico</b> <input type="text" value="adrian.garrido01@goumh.umh.es"/>	<b>Fecha</b> <input type="text" value="16/06/1998"/>
<b>Teléfono</b> <input type="text" value="666666666"/>	<b>URL</b> <input type="text" value="http://umh.es"/>
<b>C. Autónoma y Provincia</b> <div> <input type="text" value="Aragon"/> <input type="text" value="Zaragoza"/> </div>	<b>Productos de mayor interés</b> <div> <input type="text" value="ZAPATILLAS"/> <input type="text" value="ADIDAS STELLA"/> </div>
<input checked="" type="checkbox"/> Acepto los Terminos y Condiciones <input type="checkbox"/> Deseo recibir comunicaciones comerciales	

Se nos enviará automáticamente a nuestro correo los datos introducidos de la siguiente manera:



## Parte 2: contenido restringido y animación

### ▫ Categoría de noticias oculta

Para este apartado hemos utilizado dos funciones, ocultar y mostrar. Consiste básicamente en ocultar las noticias que nosotros queramos poniendo el id de nuestro div. Oculto1 y Oculto2 lo que hace es quitar las noticias visibles del menú, poniéndolas en "none", con el id dosdos también en "none", se quita el apartado de las noticias del body.

```
function ocultar() {
  document.getElementById("oculto1").style.display = "none";
  document.getElementById("oculto2").style.display = "none";
  document.getElementById("dosdos").style.display = "none";
  document.getElementById("dosdos").style.display = "none";
  document.getElementById("restringido").style.display = "block";
}

function mostrar() {
  document.getElementById("oculto1").style.display = "block";
  document.getElementById("oculto2").style.display = "block";
  document.getElementById("dosdos").style.display = "block";
  document.getElementById("dosdos").style.display = "block";
  document.getElementById("restringido").style.display = "none";
}
```

Posteriormente he optado por poner otro contenedor que se llame “restringido” para que ponga Zona Restringida, y así avisar al usuario. Por ello ponemos `display="block"`. La función mostrar es igual solo que con los displays invertidos.

Para que desde el inicio el contenido esté oculto ponemos esta línea de código en el body.

```
<body onload="ocultar()">
```

Para mostrar y volver a ocultar hemos utilizado dos imágenes como si fueran botones, y en las `img` le hemos puesto `onclick` para que al pulsar haga la función.



```


```

#### ▫ Animación con capa en movimiento

Para crear la animación me he basado en la información que tengo que dar al usuario. En este caso la información es hacer click en twitter para mostrar la noticia, y en Instagram para ocultarla.

Ambos boxes se mueven en direcciones opuestas rebotando en las paredes y pasados un tiempo corto desaparecen.



También he querido añadir otra animación de igual manera que las anteriores, pero que estuviera más tiempo para darle un toque de personalidad a la página de padel.



## Parte 3: etiquetas semánticas y RWD en HTML5

### Etiquetas semánticas y validación del W3C

En esta parte hay que cambiar los contenedores que teníamos en la práctica 1 de cabecera, menú y pie de página a expresiones de HTML 5. También tenemos que cambiar el DOCTYPE para que pase la validación del W3C.

Antes -> `<div id="header">` `<div id="menu">` `<div id="footer">`

Después -> `<header id="header">` `<nav id="menu">` `<footer id="autor">`

Doctype -> `<!DOCTYPE html>`  
`<html lang="es">`

Una vez modificado todos estos aspectos validamos y nos debería salir una etiqueta de esta forma:

**Document checking completed. No errors or warnings to show.**

### Uso de la etiqueta Picture y RWD

Para la etiqueta picture hemos empleado unas líneas de código donde le decimos que figura debe estar, y dependiendo del tamaño de la pantalla tendrá otra visualización, ya sea para móvil, PC o Tablet.

```
<picture>
  <source media="(min-width: 800px)" srcset="imagenes/instagram_new_32.png">
  
</picture>
```

Con esto se nos facilita mucho a la hora de tener un Responsive Web Design. Por este motivo es recomendable utilizar picture en vez de img ya que le puedes modificar los parámetros a tu gusto y que en cada momento tengas un icono diferente. No obstante, todo depende de la página web y del programador.

Un ejemplo de como quedaría nuestro diseño a pantalla PC o Tablet-móvil, es el siguiente:



Autor: Adrián Garrido Pantaleón  
Correo: [adrian.garrido01@goumh.umh.es](mailto:adrian.garrido01@goumh.umh.es)

Autor: Adrián Garrido Pantaleón  
Correo: [adrian.garrido01@goumh.umh.es](mailto:adrian.garrido01@goumh.umh.es)

## Valoración personal

Por último, en este apartado quiero aportar mi valoración respecto a la práctica y al trabajar tanto con HTML 5 como con JS.

Es cierto que la herramienta de HTML 5 me ha resultado muy cómoda ya que nos brinda cierta facilidad a la hora de programar y no pone tantas restricciones. También el uso de nuevos atributos y etiquetas me han hecho aprender cosas que en la primera práctica no sabía cómo iban.

Respecto a JavaScript, me ha resultado algo más difícil ya que no estaba acostumbrado al uso de esta herramienta y me ha costado más. Aun así he acabado contento con mi trabajo, a pesar de que todavía me quede recorrido para pulir ciertos aspectos.