

Desarrollo de Aplicaciones Web

Formularios Avanzado

Recuperación de Datos

Los datos introducidos mediante formularios son **enviados** al servidor. Previo envío debieran ser **validados**, para lo cual es necesaria su extracción y análisis en el lado cliente.

Los navegadores crean automáticamente un **array** denominado **forms**, conteniendo todas las referencias a formularios de la página cargada.

```
document.forms[0];
```

```
document.forms[0].elements[0];
```

Recuperación de Datos

Mecanismo más correcto → acceso a través de los atributos *name* o *id*

```
document.mi_formulario.mi_elemento;
```

o mediante los métodos funciones de DOM, para acceso directo a nodos:

```
document.getElementById("mi_elemento");
```

Recuperación de Datos

Cada elemento dispone de una serie de propiedades, que permiten el desarrollo de aplicaciones.

```
document.getElementById("mi_elemento").value.
```

Evitación de Duplicados

Permite evitar el envío de un formulario más de una vez, evitando posibles transacciones incorrectas por parte del usuario.

```
<form id="mi_formulario" action="mi_accion">  
  ....  
  <input type="button" value="Enviar" onclick="this.disabled=true;  
                                             this.value='Enviando ...';  
                                             this.form.submit()" />  
</form>
```

En lugar de.

```
<form id="mi_formulario" action="mi_accion">  
  ....  
  <input type="submit"/>  
</form>
```

Validación

Permite analizar los datos introducidos por los usuarios en el formulario, para comprobar tanto su existencia como su coherencia.

Una forma típica de hacerlo es empleando **expresiones regulares** y funciones que incluyan escaleras **if-else-if**.

```
<form id="mi_formulario" action="mi_accion" ... onsubmit="return validar()">  
....  
</form>
```

```
function validar(){  
  if (condición a cumplir por 1er campo no se cumple){  
    alert('Mensaje_1');  
    return false  
  }  
  else if (condición a cumplir por 1er campo no se cumple){  
    alert('Mensaje_1');  
    return false  
  }  
  .....  
  return true;  
}
```

Validación. HTML5.

HTML5 proporciona atributos que permiten realizar validaciones de elementos conocidos para la etiqueta <input>.

Atributo **email**.- análogo a text, pero para direcciones de email. En caso de introducir una dirección válido, solita de nuevo un valor.

Ej.-

```
e-mail: <input type="email" name="mail">  
<input type="submit" value="enviar">
```

e-mail:

Por favor, introduzca una dirección de correo.

El funcionamiento es análogo para los atributos: **url** y **number**

Validación. HTML5.

HTML5 proporciona un atributo para la etiqueta `<input>` que obliga a rellenar un elemento del formulario.

Atributos `url`, `number`,.- análogo a `text`, pero para direcciones de email. En caso de introducir una dirección válido, solita de nuevo un valor.

Ej.-

```
e-mail: <input type="email" name="mail" required>  
<input type="submit" value="enviar">
```

e-mail:

Por favor, rellene este campo.