

Formularios

Un formulario **HTML** es una sección del documento que contiene información como elementos especiales llamados controles (Casillas de verificación, radiobotones, menús, etc.) y rótulos en esos controles.

Generalmente los usuarios “completan” un formulario modificando sus controles, es decir, introduciendo información como texto, números, o seleccionando objetos en un menú.

Los controles en un formulario son elementos que interaccionan con los usuarios. Cada control tiene un valor inicial como un valor actual, que son cadenas de caracteres. El valor inicial de un control de un formulario se puede especificar mediante el atributo **value** del elemento de control. El valor inicial de un control nunca cambia, al reiniciar el formulario, el valor actual del control regresa a su valor inicial. Si un valor inicial de un formulario no está definido, al reiniciar el formulario es valor inicial queda como *indefinido*.

Tipos de controles

- Botones
- Casillas de verificación (Checkbox)
- Radio botones (Radiobutton).
- Menús (Menus)
- Entrada de texto (Text Input)
- Selección de archivos (File select)
- Controles ocultos (Hidden controls)
- Controles de tipo de objeto (Object controls)

Dirigir la atención hacia un elemento

En un documento HTML, el usuario debe dirigir la atención hacia un elemento para que éste se active y pueda realizar sus funciones. Por ejemplo, activar un enlace especificado con una etiqueta `<a>` para seguir el vínculo. De la misma forma, un

usuario debe enfocar su atención hacia un elemento text área para poder introducir texto en su interior.

Existen distintas formas de poder dirigir la atención hacia un elemento:

- Designar el elemento con un dispositivo apuntador.
- Navegar de un elemento a otro usando el teclado.
- Seleccionar el elemento mediante una tecla de acceso.

Envío de formularios

Existen algunos métodos que permiten enviar los datos de los formularios a los elementos que procesarán la información.

El atributo method del elemento FORM especifica el método HTTP usado para enviar el formulario a un agente procesador, este atributo puede tener dos valores:

- **get**: El conjunto de datos del formulario se agrega al URI especificado por el atributo action.
- **post**: El conjunto de datos del formulario se incluye en el cuerpo del formulario y se envía al agente procesador.

El método **get** debería usarse cuando el formulario no tiene efectos secundarios.

Muchas búsquedas en bases de datos no tienen efectos secundarios visibles.

El método **post** está asociado a aplicaciones con formularios que causan efectos secundarios, por ejemplo, si el formulario modifica una base de datos o la suscripción a algún servicio.

Códigos de Estado HTTP

Existe una lista de códigos de respuesta HTTP donde cada código está asociado con una frase estándar destinada a dar una descripción intuitiva del estatus.

- **Respuestas informativas (1XX)**: Petición recibida, continuando proceso.
Significa que el servidor ha recibido los encabezados de la petición y el cliente debería proceder a enviar el cuerpo de la misma.
 - **100 – continue**: El navegador puede continuar realizando su petición.

- **101 – Switching Protocols:** El servidor acepta el cambio de protocolo propuesto por el navegador.
- **102 – Processing:** El servidor está procesando la petición del navegador, pero aún no ha finalizado. Se usa para evitar que el navegador “piense” que la petición se ha perdido al no recibir alguna respuesta.
- **103 – Checkpoint:** Se va a reanudar una petición POST o PUT que fue abortada previamente.
- **Peticiones correctas (2XX):** Esos códigos indican que las peticiones fueron recibidas correctamente, entendidas y aceptadas.
 - **200 – OK:** Respuesta estándar para peticiones correctas.
 - **204 – No Content:** La petición se ha completado con éxito, pero no tiene ningún contenido.
 - **206 – Partial Content:** La petición servirá parcialmente el contenido solicitado. Esta característica es utilizada por herramientas de descarga como wget para continuar la transferencia de descargas anteriormente interrumpidas, o para dividir una descarga y procesar las partes simultáneamente.
- **Redirecciones (3XX):** El cliente tiene que tomar una acción adicional para completar la petición. La acción requerida se puede llevar a cabo por el **agente de usuario** sin la interacción del usuario sí y sólo si el método utilizado en la segunda petición es GET o HEAD. El agente de usuario no debe redirigir automáticamente una petición más de 5 veces, dado que tal funcionamiento indica usualmente un **bucle infinito**.
 - **300 – Multiple Choices:** Indica opciones múltiples para el URI que el cliente podría seguir. Esto podría ser utilizado, por ejemplo, para presentar distintas opciones de formato para vídeo, listar archivos con distintas extensiones o Word sense disambiguation.
 - **301 – Moved Permanently:** Esta y todas las peticiones futuras deben ser dirigidas a la URI dada.
 - **302 – Found:** Este es el código de redirección más popular, pero también un ejemplo de las prácticas de la industria contradiciendo el estándar.

- **305 – Use Proxy:** Muchos clientes HTTP (Como **Mozilla** e **Internet Explorer**) no se apegan al estándar al procesar respuestas con este código, principalmente con motivos de seguridad.
- **308 – Permanent Redirect:** El recurso solicitado por el navegador se encuentra en otro lugar y este cambio es permanente. A diferencia del código **301**, no se permite cambiar el método HTTP para realizar una nueva petición.
- **Errores del cliente (4XX):** La solicitud contiene sintaxis incorrecta o no puede procesarse. La intención de esta clase de códigos de respuesta es para casos en los que el cliente parece haber errado la petición. Estos códigos de estado se aplican a cualquier método de solicitud (**GET** o **POST**). Los agentes de usuario deben desplegar cualquier entidad al usuario. Éstos son típicamente los códigos de respuesta de error más comúnmente encontrados.
 - **400 – Bad Request:** La solicitud contiene sintaxis errónea y no debería repetirse.
 - **403 – Forbidden:** La solicitud fue legal, pero el servidor se rehúsa a responderla dado que el cliente no tiene los privilegios para hacerla.
 - **404 – Not Found:** Recurso no encontrado. Se utiliza cuando el servidor web no encuentra la página o recurso solicitado.
 - **408 – Request Timeout:** El cliente falló al continuar la petición, excepto durante la ejecución de vídeos Adobe Flash cuando sólo significa que el usuario cerró la ventana de vídeo o se movió a otro.
 - **415 – Unsupported Media Type:** La petición del navegador tiene un formato que no entiende el servidor y por eso no se procesa.
 - **418 – I'm a Teapot:** Soy una tetera.
 - **429 – Too Many Requests:** Hay muchas conexiones desde esta dirección de Internet.
 - **451 – Unavailable for Legal Reasons:** El contenido ha sido eliminado como consecuencia de una orden judicial o sentencia emitida por un tribunal.

- **Errores de servidor (5XX):** El servidor falló al completar una solicitud aparentemente válida. Estos códigos de respuesta indican casos en los cuales el servidor tiene registrado aún antes de servir una solicitud, que está errado o es incapaz de ejecutar la petición. Estos códigos de repuesta son aplicables a cualquier método de petición.
 - **500 – Internal Server Error:** Es un código comúnmente emitido por aplicaciones empotradas en servidores web, mismas que generan contenido dinámicamente, por ejemplo, aplicaciones montadas en **IIS** o **Tomcat**, cuando se encuentran con situaciones de error ajenas a la naturaleza del servidor web.
 - **502 – Bad Gateway:** El servidor está actuando de proxy o Gateway y ha recibido una respuesta inválida del otro servidor, por lo que no puede responder adecuadamente a la petición del navegador.
 - **504 – Gateway Timeout:** Similar a la descripción del mensaje de estado 502.
 - **511 – Network Authentication Required:** El navegador debe autenticarse para poder realizar peticiones, por ejemplo, con portales cautivos que te obligan a autenticarte antes de empezar a navegar.