SEMANA 1

DOCUMENTO 1

DOCUMENTO 1

Formulario

Un Formulario es un elemento que permite recoger datos introducidos por el usuario. Los datos serán procesados por el programa que indiquemos. Los Formularios se utilizan para obtener distintos tipos de información, por ejemplo, los datos de una compra, inscribirse en páginas, dar opiniones en sitio web, etc.

Los Formularios son una excelente herramienta para comunicarse con los usuarios pero implican unos riegos en Seguridad que hay que tener en cuenta Ya que son el camino más usados para ataques malicioso.

La Etiqueta <Form>: Aunque un formulario utiliza la etiqueta form como parte central, necesita de otras etiquetas para poder funcionar, normalmente

incluye las etiquetas input y label, y opcionalmente otras más. Es decir, un formulario es un componente de una página web que está formado por varios elementos, como campos de texto, casillas de verificación, opciones desplegables, botones y otras etiquetas para darle formato.

Todos los elementos del formulario están dentro de las etiquetas <form> y </form>. por ejemplo, el siguiente código muestra un formulario básico:

```
<form action="#" method="post">
<p> Escribe aquí tu petición <input type="text"
name="mensaje"> </p>
<p> <input type="submit" value="Enviar">
</p>
</form>
```

Produce este resultado:
Escribe aquí tu petición

Enviar

El formulario empieza con la etiqueta <form> y acaba con la etiqueta </form>. Entre estas dos etiquetas podemos escribir las etiquetas que queramos

para declarar los campos de entrada o controles del formulario mediante la etiqueta <input>, también podemos escribir etiquetas de html para describir cada campo del formulario, en este caso hemos utilizado etiquetas <p> aunque es más apropiado utilizar etiquetas label, como veremos más adelante.

Dentro de la etiqueta <Form> ponemos los atributos que necesitamos, normalmente el atributo action para indicar el archivo que tratará los datos y el atributo method con el método para enviar los datos.

También hay que poner un botón para enviar el formulario.

Atributos de la Etiqueta <Form>

El atributo action indica una URL (dirección) a la que enviar los datos del formulario. Esta URL debe ser un elemento capaz de recibir los datos del formulario y procesarlos. Normalmente es un programa escrito en un lenguaje de programación web (PHP, Java, ...) que se ejecutará en el servidor web. por ejemplo, el programa puede realizar una validación de los datos, grabarlos en

en una base de datos Y arrancar otro programa para continuar con el proceso, o devolver el control al usuario para que continué con el proceso. En los casos en que los datos no requieren un tratamiento complejo también se puede utilizar un lenguaje de programación que se ejecute en el navegador web (JavaScript). Si no se especifica el atributo action, o se escriba action="#" los datos los recibe la misma página que contiene el formulario.

Al recibir datos de un usuario a través de un formulario es conveniente Validar los datos, esto se puede hacer de diversas formas. Vamos a ver las tres más comunes.

1. Validar con el propio formulario. La primera validación se suele hacer con los mecanismos que dispone el Formulario para realizar una Validación Sencilla, por ejemplo, comprobar que una fecha es correcta o que un dato no se ha dejado en blanco, esto lo veremos más adelante.

2. Validar en el navegador. Al pulsar en el botón para enviar, el Formulario se puede llamar a un programa que realice una validación en el navegador web, esta manera de validar se

realiza cuando no es posible realizarla con los mecanismos del Formulario, por ejemplo, comprobar que un NIF es correcto.

3. Validar en el Servidor. Se pueden validar los datos desde el lado del servidor web con programación web, esto se hace cuando las validaciones necesitan acceder a una base de dato, por ejemplo, para saber si un usuario ya está registrado en un sitio web o es un alta nueva.

El atributo method indica el método mediante el que se transfieren las variables del formulario. Su valor puede ser get o post. El método es una caractirística del protocolo HTTP y afecta a la forma de comunicación, no al contenido de los datos.

Con valor get los datos se envian junto con la URL del Formulario por lo cual cualquiera puede verlos en la barra de direcciones del navegador. También quedan almacenados en la historia del navegador. Se considera un método poco seguro, por este método no se deben enviar passwords, ni datos sensibles. Es el método que se usa cuando no se especifica el atributo method.

Con el valor post los datos se envían de forma interna, no junto a la URL, no son visibles en la barra de direcciones del navegador, ni se almacenan en la historial del navegador. por lo que se considera un método más seguro que el método get.

El atributo autocomplete admite dos valores, on y off. Si escribimos autocomplete="on", el navegador intentará mostrar sugerencias al usuario para rellenar los datos según lo introducido previamente en otros formularios. por ejemplo, si hay un campo que se llama apellidos y otras veces ya lo hemos rellenado, el navegador nos mostrará una lista con apellidos por si queremos elegir alguno de ellos. Si autocomplete="off", no se mostrarán sugerencias podemos especificar este atributo a nivel de la etiqueta form, con lo cual afectará a todos los campos, o a nivel de cada campo, en este caso tendrá preferencia lo indicado a nivel de campo.

El atributo novalidate se aparece, indica que no se realicen las validaciones indicadas a nivel de los campos del formulario. por ejemplo, si hemos indicado que un campo

al número Y en la siguiente [illegible] especificados. novalidate se permitirán datos no numéricos en ese campo [illegible] tipo submit disponemos del atributo formnovalidate para indicar que si se pulsa ese botón no se validen los campos del formulario.

El atributo actype indica el modo en que será cifrada la información para su envío. por defecto tiene el valor application/x-www-form-urlencoded.

En el siguiente formulario vamos a ver algunos ejemplos de estos atributos, el formulario llamará al programa recibir.php para tratar los datos, con el método get y el tipo de encriptación application/x-www-form-urlencoded, no se realizarán validaciones de datos y el campo de entrada no mostrará sugerencias para rellenarlo.

```
<form action="recibir.php" method="get" novalidate
enctype="application/x-www-form-ulencoded">
<p> escribe aqui tu petición <input type="text"
name="mensaje" autocomplete="Off"> </p>
<p> <input type="Submit" Value="Enviar">
</p>
</Form>
```

SEMANA 2

DOCUMENTO 2

DOCUMENTO #2

ETIQUETA <INPUT> DE UN FORMULARIO

La etiqueta input es la encargada de crear los campos de entrada del formulario. Aunque la etiqueta Input también se puede utilizar sin estar ligada a un formulario. Existen bastantes tipos de datos para el elemento Input, por ejemplo, texto, casilla de verificación, área de texto, fecha, etc., para cada uno de ellos el navegador web crea un campo con el formato adecuado de forma automática. El atributo type define el tipo del elemento input.

Para cada tipo de input podemos añadir otros atributos para controlar diferentes características. Hay atributos genéricos que se pueden incluir en la mayoría de tipos de campo de entrada, hay otros atributos más específicos que solo se pueden incluir en unos tipos concretos. por ejemplo para todos los tipos podemos incluir el atributo name que identifica el campo de entrada, pero el atributo checked sólo tiene sentido con los tipos de campo

Checkbox Y radio, como veremos más adelante.

Los atributos más comunes son name, value, required, placeholder, size, maxlenght, minlgth, autofocus, disabled, sic Y alt. La función de cada uno de estos atributos la vamos a ver a continuación. Según vamos explicando los diferentes tipos de campos, de la misma forma iremos viendo los atributos específicos para cada caso.

Tipo text:
Define un campo de texto, es el atributo más usado Y se usa para introducir una linea de texto. Se puede introducir cualquier carácter, letras, números, etc. Los navegadores lo presentan como una caja de texto una longitud de 20 caracteres. Esta longitud se puede variar con el atributo size, este valor se refiere a la longitud de visualización de la caja, pero el usuario puede introducir más caracteres; si queremos controlar esa cantidad podemos utilizar los atributos maxlegth Y minlength que indican el máximo Y mínimo de caracteres que puede introducir el usuario.

por ejemplo, en el siguiente formulario el campo nombre tiene la longitud por defecto pero el usuario puede introducir más caracteres; el campo

apellidos tiene una longitud de 15 caracteres Y hemos limitado a 15 los caracteres que puede introducir; el campo localidad tiene una longitud de 18 Y como mínimo el usuario debe introducir 2 caracteres.

```
<form action="#" method="post">
<p>Nombre:<input type="text" name="Nombre"></p>
<p>Apellidos:<input type="text" name="ape" size="15" maxlength="15"></p>
<p>Localidad:<input type="text" name="localidad" size="18" minlength "2"></p>
<p><input type="submit" Value="Enviar"></p>
</form>
```

Nombre:

Apellidos:

Localidad

Enviar

Si el usuario introduce un número de caracteres que no cumple con las condiciones de maxlength y minlength, aparecerá de forma automática un globo indicándole el error, como muestra la siguiente imagen para el caso de introducir menos de dos caracteres en el campo localidad.

Un atributo importantes es name que permite dar un nombre diferentes a cada campo, de esta forma el programa que reciba los datos podrá identificar cada campo, en el ejemplo anterior al campos para introducir los apellidos le hemos dado el valor ape en el atributo name.

Otro atributo muy utilizado es value que permite establecer un valor por derecto. por ejemplo, en el campo localidad hemos escrito value="Valencia" de forma que el texto Valencia aparecerá cuando se cargue el formulario podrá dejar este valor o modificarlo.

para mostrar información dentro de un campo disponemos del atributo placeholder que muestra un texto al cargarse el formulario, pero a diferencia del atributo value, estos datos no se envían, son sólo informativos. Los datos se borran cuando el usuario coloca el cursor en el campo.

El atributo required hace que sea obligatorio rellenar ese campo, es decir, no se puede dejar en blanco.

En el siguiente ejemplo podeis ver el comportamiento diferente de los atributos Value y placeholder; así como comprobar que si no rellenas el campo nombre aparecerá un globo indicando que debes completar ese campo, puesto que tiene el atributo required.

```
<form action="#" method="post">
<p>Nombre: <input type="text" required name="nombre">
</p>
<p>Apellidos: <input type="text" placeholder="Escriba
los dos apellidos" name="Ape" size="25"></p>
<p> Localidad: <input type="text" Value="Valencia"
name="localidad" size="18" minlength="2"></p>
<p><input type="Submit" Value="Enviar"></p>
</form>
```

Nombre:

Apellidos: Escriba los dos apellidos

Localidad: Valencia

Enviar

También podemos limitar los valores que permitimos mediante una lista desplegable, para ello debemos utilizar el atributo list con un nombre de lista, luego debemos crear la lista con la etiqueta datalist, y crear los valores de la lista con la etiqueta option, tal y como muestra el siguiente código:

```
<p> Elegir memoria USB: <input list="listausb"></p>
<datalist id="listausb">
<option value="8">
<option value="16">
<option value="32">
<option value="64">
</datalist>
```

Este atributo list se puede utilizar en otros muchos campos como text, date.

SEMANA 3

DOCUMENTO 3

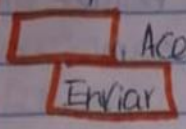
DOCUMENTO #3

TIPO DE CHECKBOX

Define una casilla de verificación que puede tomar dos valores: marca o desmarca. Por esto se utiliza para datos que puedan tomar dos valores, si/no de acuerdo/en desacuerdo, activado/desactivado. por ejemplo se puede utilizar en estos casos: Nacionalidad española, Aceptar condiciones, pantalla completa, etc. Si añadimos el atributo checked la casilla aparecera marca inicialmente.

por ejemplo, en este código aceptamos o no las condiciones que se nos proponen en otra parte de la página por defecto aparecerá marcada, el atributo name identifica el campo de igual forma que ya hemos visto anteriormente, Sin embargo el atributo Value tiene un significado un poco diferente al caso de un campo de texto, puesto que aquí el usuario no puede introducir ningún valor, el valor lo establecemos nosotros en el atributo value, si el usuario marca la casilla se enviará este valor (en este ejemplo: acepta) Y en caso contrario se enviará el valor en blanco.

<Input tipo="checkbox" value "acepta" name= "condiciones" checked="checked"> Acepto condiciones </p>



TIPO RADIO

Con este tipo de entrada se define una opción a elegir. Normalmente se presentan varias opciones sobre una cuestión para que el usuario elija sólo una de ellas. Por ejemplo, sexo: hombre o mujer; tipo domicilio: calle, plaza, avenida; Transporte: avión, tren, autobús. Se deben escribir tantas etiquetas Input como opciones necesitamos, y todas deben tener el mismo valor en el atributo name, de esta forma se sabe que están relacionadas y que sólo se puede elegir una. Además el atributo Value debe tener valores distintos para que el programa que recibe los datos sepa cuál de las opciones ha sido elegida.

Por ejemplo, en el siguiente código presentamos dos opciones para elegir el sexo; en las dos etiquetas input el valor del atributo name es sexo y en un caso el atributo value es h y el otro m.

Si quisiéramos que una opción apareciese marcada inicialmente podríamos utilizar el atributo checked.

Sexo: Hombre <input type="radio" name="sexo" value="h">Mujer <input type="radio" name="sexo" value="m">

Sexo: Hombre ○ Mujer ○

Enviar

TIPO NUMBER

Sirve para introducir números. Por defecto sólo permite introducir números enteros (números sin decimales, positivos y negativos) para con el atributo step podemos permitir los decimales que queramos, por ejemplo con step=0.1 permitimos un decimal con step=0.01 dos decimales. En realidad el atributo step establece el incremento que puede haber entre un valor y el siguiente, por ejemplo, si escribimos step=10 sólo estamos permitiendo múltiplos de 10. Los navegadores suelen mostrar dos pequeñas flechas en la parte derecha del campo de entrada para ir disminuyendo o aumentando el número

en la cantidad indicada por el atributo step, si no indicamos el atributo step, toma el valor 1. Según el navegador y el idioma se admite la coma decimal, el punto decimal, o ambos.

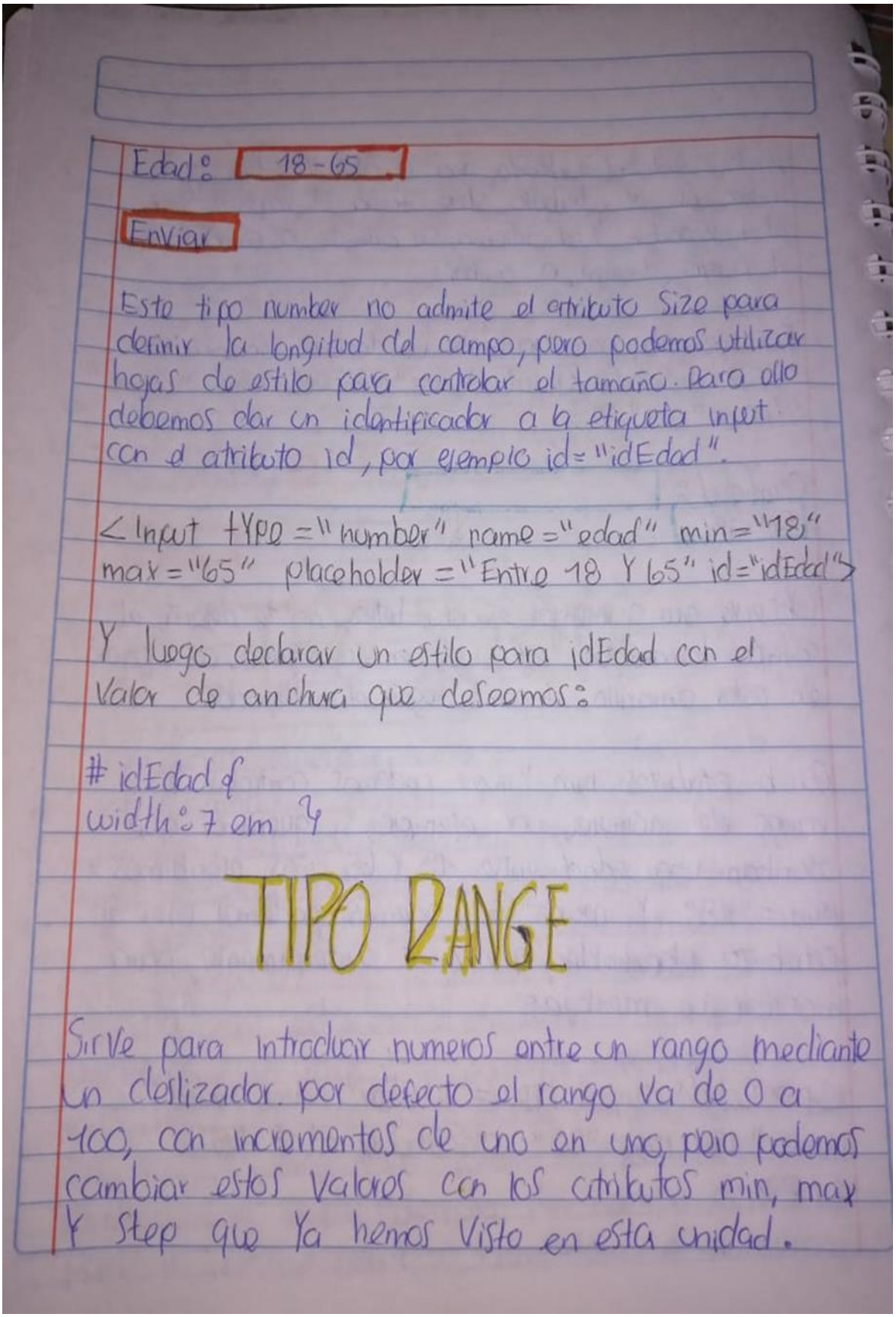
por ejemplo, para permitir un decimal escribimos:
 Cantidad: <input type="number" name="número" step="0.1" /p>

Cantidad:
Enviar

Observa que si intentas escribir letras, no te dejará, el campo tomará un valor rojo en su marco, o se pondrá en color amarillo el fondo, según el navegador.

Con los atributos min y max podemos controlar el rango del número, por ejemplo, si queremos que escriban una edad entre 18 y 65 años, escribimos: min="18" y max="65", además podemos usar el atributo placeholder, que vimos anteriormente, para mostrar un mensaje:

 Edad: <input type="number" name="edad" min="18" max="65" placeholder="18-65"


Edad: 18-65
Enviar
Este tipo number no admite el atributo Size para definir la longitud del campo, pero podemos utilizar hojas de estilo para controlar el tamaño. Para ello debemos dar un identificador a la etiqueta input con el atributo id, por ejemplo id="idEdad".
<Input tYpe="number" name="edad" min="18" max="65" placeholder="Entre 18 Y 65" id="idEdad">
Y luego declarar un estilo para idEdad con el valor de anchura que deseemos:
#idEdad{
width: 7em }
TIPO RANGE
Sirve para introducir numeros entre un rango mediante un deslizador. por defecto el rango va de 0 a 100, con incrementos de uno en uno, pero podemos cambiar estos valores con los atributos min, max y step que ya hemos visto en esta unidad.

Introducir Valores con el deslizador puede ser un poco impreciso si el rango es amplio por lo que suele utilizarse para introducir Valores aproximados, como la Valoración de un producto, o para rangos pequeños, El Formato del mando deslizador Varía ligeramente Según cada navegador web.

por ejemplo, el Siguiente código pide Valorar un producto entre 10 Y 100 con incrementos de 5:

Volumen (10-100): [redacted]

Enviar

TIPO PASSWORD

Sirve para introducir claves Y contraseñas, la diferencia principal respecto a un campo de texto es que cuando el usuario está escribiendo no se ven los caracteres para evitar que alguien pueda copiarlos. Un ejemplo Sencillo es el Siguiente:

 Contraseña: <Input type="password" name="clave" >

Contraseña:

Enviar

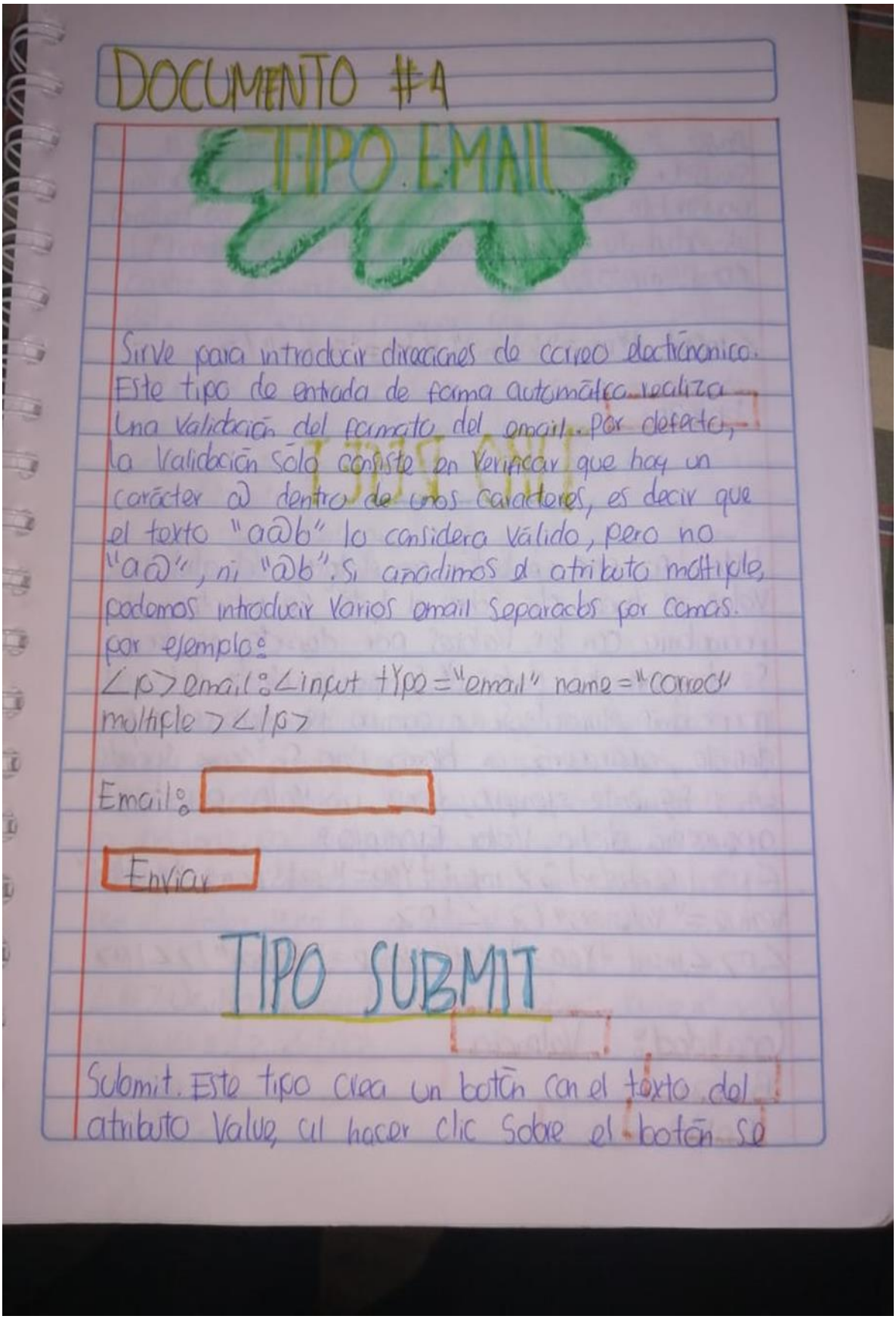
Vamos a aprovechar para explicar un atributo genérico que suele colocar en el primer campo de un formulario, se trata del atributo autofocus. Cuando se carga un formulario el foco o cursor no se sitúa en ningún campo del formulario por lo que el usuario debe hacer clic en el campo que desea rellenar (normalmente el primer campo) antes de poder escribir nada, Si queremos evitarle este clic inicial podemos añadir el atributo autofocus, de esta forma el cursor irá directamente a ese campo. En este ejemplo puedes comprobar cómo, al cargarse la página, el cursor se coloca en el campo Contraseña, mientras que en el formulario anterior no aparecerá el cursor:

Contraseña:<input type="password" name="clave" autofocus>

También se puede añadir el atributo inputmode que permite que el dispositivo muestre un modo de entrada (teclado) distintos del estándar. Esto puede ser especialmente interesante si queremos introducir números ya que el poner inputmode="numeric" se debe abrir el teclado numérico en los teléfonos móviles. por defecto el inputmode tiene el valor Verbatin para caracteres alfanuméricos.

SEMANA 4

DOCUMENTO 4


DOCUMENTO #4
TIPO EMAIL
Sirve para introducir direcciones de correo electrónico. Este tipo de entrada de forma automática realiza una validación del formato del email. Por defecto, la validación sólo consiste en verificar que hay un carácter @ dentro de unos caracteres, es decir que el texto "a@b" lo considera válido, pero no "a@", ni "@b". Si añadimos el atributo multiple, podemos introducir varios email separados por comas. por ejemplo:
email:<input type="email" name="correo" multiple>
Email:
Enviar
TIPO SUBMIT
Submit. Este tipo crea un botón con el texto del atributo Value, al hacer clic sobre el botón se

SE

envía el formulario, es decir, se ejecutara en el servidor el programa indicado en el atributo action pasándole los valores de los campos. Si no podemos el atributo Value aparecerá el texto "Submit" por ejemplo:

<Input type="Submit" Value="EnViar" />

EnViar

TIPO RESET

Este tipo crea un botón con el texto del atributo Value, al hacer clic sobre el botón se inicializa el formulario con los valores por defecto, es decir se borran los datos y se parte desde del formulario inicial: Si un campo no tiene valor por defecto, aparecerá en blanco, pero si, como sucede en el siguiente ejemplo, tiene un valor por defecto, aparecerá dicho valor. Ejemplo:

 Localidad: <Input type="text" name="ciudad" Value="Valencia" />
 <Input type="reset" Value="Borrar" />

Localidad: Valencia
Borrar
EnViar

TIPO HIDDEN

Este tipo de input se utiliza para enviar información oculta en el formulario, el usuario no verá el contenido del campo pero el programa que recibe los datos sí recibirá el dato asignado con el atributo Value. La utilidad de este atributo puede ser muy variada y depende de las necesidades del programador. Por ejemplo, si recibamos pedidos desde dos páginas web diferentes podemos escribir en un campo oculto un valor que identifique cada página para después saber de qué página ha venido un determinado pedido.

No es conveniente emplear este campo oculto para temas de seguridad ya que aunque no se muestra al usuario, este puede verlo si pulsa el botón secundario del ratón y elige "ver código fuente de la página". Por ejemplo, en este formulario hemos escrito un campo de texto y un campo oculto con el valor "no se puede ver":

<P> Oculto: <input type="hidden" value="no se puede ver" /> </P>

Nombre:

Oculto:

Enviar

TIPO DATE

Este tipo de Input para introducir fechas (años, mes y día) son escribibles en el campo o bien mediante una pequeña Ventana emergente donde se puede hacer clic en una fecha. Esta Ventana tiene diferentes aspecto según cada navegador pero la función es la misma y permite elegir una fecha haciendo clic con el ratón. Por ejemplo, en Chrome se muestran unas pequeñas flechas para aumentar o disminuir el valor, y una flecha más grande a la derecha para abrir la Ventana emergente.

Inicialmente, en el campo se muestra el formato permitido de la fecha, en español normalmente es dd/mm/aaaa, para que el usuario sepa cómo tiene que introducir la fecha. El propio navegador suele realizar la validación de la fecha e impide introducir fechas incorrectas, por ejemplo el mes 13 o día 45.

Podemos inicializar el campo con una fecha concreta con el atributo value, pero hay que tener en cuenta que el formato interno de value y del propio campo es siempre "aaaa-mm-dd" mientras que el formato de

Visualización y entrada depende de la configuración del navegador y Sistema Operativo, como hemos dicho en español es dd/mm/aaaa. Es decir que aunque escribamos y veamos 30/12/2018, el contenido del campo que se pasará al programa que recibe los datos será 2018-12-30. Ejemplo:

```
<Input type="date" name="fecha" value="2020-06-23" />
```

Fecha: 23/05/2020
Enviar

TIPO TIME

Con este tipo de entrada podemos introducir un tiempo, por defecto, horas y minutos, y opcionalmente segundos. El formato de visualización es hh:min, con rango de horas de 023, aunque algunos navegadores pueden mostrar las horas como 0-11 pm/am. El propio navegador suele realizar la validación de la hora e impide introducir horas incorrectas, por ejemplo la hora 25 o el minuto 61.

```
<Input type="time name="hordnicio"
value="23:30" />
```