

ISSD

# I2

—Asc—  
Analista de  
Sistemas

## Informática 2

Módulo didáctico - 2015

FORMULARIOS,  
CONTROLES Y  
SECTORES



Unidad 1 y 2



Clase 8



(U1)HTML  
(U2) CSS



- | Usar e implementar selectores para formularios.
- | Entender la importancia de aplicar estilos a los controles de formularios.
- | Conocer la importancia de validar el ingreso de datos mediante controles sin necesidad de otro lenguaje o script.
- | Comprender los beneficios de maquetar los formularios con CSS3.
- | Entender los tópicos expuestos en la clase a través de ejercicios comentados.



¿No te pasó que al ver un sitio poco colorido o poco profesional, no te dan ganas de realizar ninguna transacción en él, o te causa simplemente aburrimiento tener que usar una interfaz poco atractiva?

Si esto te afecta, no dejes de leer esta interesante clase. En ella encontrarás gran variedad de estilos para mejorar el aspecto de tus formularios y así poder trabajarlos a nivel profesional. Suerte con la lectura, ¡nos vemos en la próxima clase!

### CONTACTO

	Nombre
	Su e-mail
	Asunto
	Escriba su mensaje...

 **ENVIAR**

### Formulario de Registro

Nombre (requerido)

Email (requerido)

Contraseña (requerido)

Nickname (requerido)

**Regístrame!!**

# *Formularios, controles y selectores*

En esta clase aprenderemos tres tópicos centrales:

- | Mejoraremos el aspecto de los formularios con CSS3,
- | Trabajaremos con los controles propiamente dichos,
- | y con selectores para validar la carga de datos.

Tambien hoy encontrarás un monton de ejercicios ejemplificando el contenido de la clase. Espero que el material que te proponemos cumpla tus expectativas, te deseo una lectura entretenida y un buen desarrollo de ejercicios.

¡Mucha suerte!





## *Desempeño de Investifación*

Se exponen diseños de formularios adicionales para su codificación.  
Se presentan distintos casos de ejemplos y referencias para que vos  
perfiles el desempeño de síntesis final.



# Selectores para Formularios

## Pseudo clase :valid e :invalid

Básicamente la pseudo clase: invalid representa cualquier campo invalido que sea resultado de una validación (valga la redundancia), lo opuesto ocurre con su contraparte: valid. Esto facilita que si un campo contiene un valor inválido pueda adoptar una apariencia acorde, de esta forma ayuda al usuario a identificar los campos que necesitan ser verificados.

Estos selectores funcionan a través del atributo HTML required. El selector :invalid funciona con la presencia de este atributo en el elemento HTML, el cual significa que ese campo de formulario debe ser completado

para ser válido y enviado. Esto es una forma de aplicarle reglas a determinados estados de los campos de formularios. Su contraparte :valid funciona cuando no se encuentra definido el atributo required en el elemento HTML y por consecuencia el campo será siempre válido.

### HTML

```
<form>
< label for="nombre">Nombre</label>
< input type="text" id="nombre" required/>
< label for="apellido">Apellido</label>
< input type="text" id="apellido"
required/><br/>
< label for="email">E-mail</label>
< input type="email" id="email" required/><br/>
< label for="direccion">Direcci&oacute;n;</
```

label>

```
< input type="text" id="direccion"/><br/>
< label for="nacionalidad">Tel&eacute;fono</
label>
< input type="text" id="telefono"/><br/>
< /form>
```

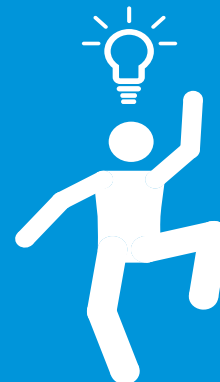
### CSS

```
input:valid{background-
color:rgba(0,204,153,0.3);}
input:invalid{background-
color:rgba(255,51,0,0.3);}
```

El formulario cuenta con 5 campos de texto; los tres primeros tienen el atributo required, por ende reaccionarán al selector **:invalid** teniendo, como lo dicen las reglas, un color rosado. Al colocar algún texto en cualquiera de los 3 primeros, el color cambiará a verde pastel



Un aspecto que es importante resaltar, es que por defecto todos los campos son validos –como es de esperarse –, pero si un campo es requerido, entonces ocurre lo contrario, este por defecto será inválido.



```
input:required:invalid  
{border: 1px solid red;}
```

```
input:required:valid  
{border: 1px solid green;}
```

siendo en ese momento válido el campo de texto y por consecuencia lo afectara el selector **:valid**. Otro ejemplo:

```
input:invalid {border: 1px solid red;}  
input:valid {border: 1px solid green; }
```

## Pseudo clase Required

Esta pseudo clase nos permite definir el estilo para el control de formulario cuando es obligatorio ingresar un dato.

```
[required]{ border-color: green; box-shadow:  
0px 0px 5px green;}
```

## Pseudo clase Optional

Esta pseudo clase se aplica a los elementos **input** cuando el atributo **required** no ha sido definido, es decir, cuando los controles no son obligatorios. Al igual que la pseudo clase Valid, no es muy utilizada pero no esta de mas saber como usarla.

```
:optional{ border-color: yellow;}
```

## read-only, :enabled y :disabled

**:enabled** y **:disabled** son dos opuestos; los mismos reaccionan en base al atributo **HTML disabled** que determina si el elemento

HTML esta **desactivado** para ser utilizado por el usuario o activado. Este concepto de desactivado significa que el usuario no podrá escribir en el campo si **disabled** esta presente en el elemento HTML. Entonces se pueden crear unas reglas visuales para ver como se verán en ambos estados.

Por otro lado también en el siguiente ejemplo vas a ver el funcionamiento de **:read-only**, que como todos estos selectores, se basa en la presencia de un atributo HTML; en este caso de **readonly** (solo lectura). Este atributo desactiva automáticamente la posibilidad del que usuario cambie el texto del campo. Puede ser útil cuando, una vez enviado el formulario, los campos queden completados pero estáticos sin poder ser modificados.



## HTML

```

<div id="ContenedorPadre">
< form>
< label for="nombre">Nombre</label>
< input type="text" id="nombre" required/>
< label for="apellido">Apellido</label>
< input type="text" id="apellido"
required/><br/>
< label for="email">E-mail</label>
< input type="email" id="email" required/><br/>
< label for="direccion">Direcci&ocute;n</
label>
< input type="text" id="direccion"/><br/>
< label for="nacionalidad">Tel&eacute;fono</
label>
< input type="text" id="telefono"/><br/>
< label for="vivo">Vivo en</label>
< input type="text" id="vivo" readonly
value="Argentina"/><br/>
< label for="extra">No, vivo en</label>
< input type="text" id="extra" disabled/><br/>
< textarea id="textos" required></textarea>
< /form>
< /div>

```

## CSS

```

:disabled{outline: 3px solid red;}
:enabled{outline: 3px solid deepskyblue;}
:read-only{background-color: rgb(0,204,102);}

```

Como podés observar en la figura 1, los campos de formulario afectados por **:enabled** tienen un borde celeste, en ellos el usuario puede escribir. El que tiene el atributo HTML **disabled** y sera afectado por el selector **:disabled**, tendrá un borde rojo. Por último el campo que sea de solo lectura (tenga el atributo **readonly**) tendrá un fondo de color verde afectado por el selector **:read-only**.

En la imagen 2 te mostramos un formulario con dos campos de inserción de datos: el primer campo, que permite insertar el nombre, está habilitado (enabled); el segundo campo, que sirve para insertar la edad, está deshabilitado (disabled). Luego tenemos dos botones de opción para el sexo (hombre o mujer). Y por último, una casilla de selección.

Vamos ahora a diferenciar visualmente los elementos del formulario en función del estado: habilitado, deshabilitado o seleccionado.

Este es el código HTML del formulario:

```

<form id="form1" method="#" action="#">
<p> <label for="nombre">Su nombre y
apellidos: </label>

```

Figure 1 shows a form with the following elements:

- Nombre: Text input with a light blue border.
- Apellido: Text input with a light blue border.
- E-mail: Text input with a light blue border.
- Dirección: Text input with a light blue border.
- Teléfono: Text input with a light blue border.
- Vivo en: Text input with a light green background, containing the text "Argentina".
- No, vivo en: Text input with a red border.

Figure 2 shows a form with the following elements:

- Su nombre y apellidos: Text input with a light green background, containing the text "Alejandro".
- Su edad: Text input with a red border.
- hombre: Radio button.
- mujer: Radio button.
- Acepto las condiciones de uso: Checked checkbox.

```

<input type="text" id="nombre" /> </p>
<p> <label for="edad">Su edad: </label>
<input type="text" id="edad"
disabled="disabled"/> </p>
<p> <input type="radio" name="sexo"
id="hombre" value="hombre" />
<label for="hombre">hombre</label><br/>
<input type="radio" name="sexo" id="mujer"
value="mujer" />
<label for="mujer">mujer</label> </p>
<p> <input type="checkbox" id="acepto" />
<label for="acepto">Acepto las condiciones de
uso.</label> </p>
</form>

```

Y estos son los estilos CSS:

```

#acepto:checked+label {background-color:
gold;
font-weight: bold; }
#hombre:checked+label,
#mujer:checked+label {
font-style: italic; }
.enabled { background-color: lightgreen;}
.disabled {background-color: lightcoral; }

```

El primer selector se dirige a la etiqueta (label) de la casilla de selección (#acepto) y se

aplicará cuando esta aparezca seleccionada (checked). El segundo selector se dirige al botón de opción seleccionado. Los otros dos estilos añaden, simplemente, un fondo de color en función del estado del campo.

Existe otro estado para los botones de opción y las casillas de selección, se trata del estado intermedio: **:indeterminate**. Este estado indica que el elemento no está ni seleccionado, ni no seleccionado. Los campos obligatorios, que requieren que se haya insertado un valor, deben presentar el atributo booleano required. Los que no sean obligatorios, serán por defecto facultativos, sin que haga falta precisarlo.

Veamos un ejemplo de formulario: (1)

La pseudo-clase **:focus** permite, tal como se observa en la figura 2, resaltar el campo que se esté usando en ese momento con un borde **outline**. El estilo **outline-offset**, corresponde a la distancia entre el límite de la caja y el límite del borde. Veamos un ejemplo muy sencillo:

```

input:focus {outline: solid 3px green; outline-
offset: 2px; }

```

```

<form id="form1" method="#" action="#">
<p> <label for="nombre">Su nombre y
apellidos: </label>
<input type="text" name="nombre"
id="nombre" required /> </p>
<p> <label for="edad">Su edad: </label>
<input type="text" name="edad" id="edad"
/> </p>
<p> <input type="submit" id="enviar"
value="Enviar" /></p>
</form>

```

Y estos son los estilos CSS:

```

input:required { background-color: lightcoral;
}
input:optional { background-color:
lightgoldenrodyellow;}
input#enviar {background: none;}

```



# Mejorando el aspecto de formularios con CSS

Los formularios son parte importante de los sitios, pero muchas veces poco elegantes.

**CSS** hace más fácil la creación de formularios atractivos rápidamente, porque una vez que se ha desarrollado un diseño, lo podemos reproducir en cada formulario del sitio.

Usando **CSS**, podemos cambiar muchas cosas sobre el aspecto de nuestros formularios y los elementos dentro de ellos; podemos redefinir la etiqueta de **HTML** que controla estos elementos, o podemos aplicar las llamadas clases css a estas etiquetas.

## Mejorar el aspecto de algunos controles

### Personalizando el textarea

Trabajaremos con un **textarea** al que vamos a poner el fondo transparente y en el que vamos a modificar los colores de las barras de desplazamiento.

```
<textarea class=estilotextarea4 cols="30" rows="5"></textarea>
```

Para definir el estilo hemos utilizado el siguiente CSS:

```
.estilotextarea4 {  
background-color: transparent;border: 1px  
solid #000000;  
scrollbar-arrow-color: #000066;  
scrollbar-base-color: #000033;  
scrollbar-dark-shadow-color: #336699;  
scrollbar-track-color: #666633;  
scrollbar-face-color: #cc9933;  
scrollbar-shadow-color: #DDDDDD;  
scrollbar-highlight-color: #CCCCCC;}
```

El color de fondo es transparente por el atributo background-color: **transparent**; esto quiere decir, que el color del textarea tiene el mismo color que el fondo de la página donde está colocado. Si al textarea lo tenemos colocado sobre un fondo blanco, no observaremos ninguna diferencia con

respecto a otros, pero si lo tenemos sobre fondo de otro color, se verá de ese mismo color. Luego, se ha aplicado un borde de un píxel negro y con los restantes atributos se modifica el color de las barras de desplazamiento del textarea. El resultado se puede ver en la imagen 1.

## Controles con imágenes

Si estas aburrido de trabajar siempre con las mismos inputs o cajas de texto, te presento una alternativa. Utilizando un poco de diseño y css podrías reemplazar por completo esos formularios de contacto. La técnica consiste primero en ocultar el input mediante css, luego diseñamos con la aplicación de diseño gráfico que más te guste una caja de texto, como por ejemplo, la de la figura 2.

Como primer paso insertamos la caja de texto, asignándole un "id" para luego identificarlo con css, de esta manera:

```
<input id="inp1" maxlength="25" />
```

Ahora lo trabajamos un poco con CSS, donde primero asignamos la imagen de fondo que

tendrá nuestra caja con **background-image**, luego decimos que nuestro fondo no se repita y lo hacemos con **background-repeat:no-repeat**, por último damos los detalles del borde y el tamaño horizontal y vertical, que por lo general debería coincidir con el tamaño de la imagen de fondo:

```
#inp1{ border: none; width: 205px; height: 32px; padding-left: 3px; padding-top: 6px; background-image:url(input1.gif) no-repeat; }
```

Cuando el foco se posicione en el input:

```
input:focus{ background-color: transparent; }
```

De esta manera podrías diseñar no sólo cajas de texto, sino también textarea, botones, etc.

## Mejorar el aspecto de una lista

Para mejorar el aspecto de una lista, como la que te mostramos en la imagen 3, podés utilizar el siguiente código:

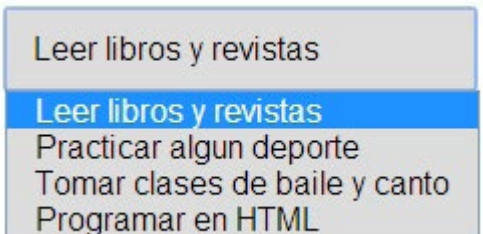
1



2



3





```
<select name="" id="">
  <option>Leer libros y revistas</option>
  <option>Practicar algun deporte</
option>
  <option>Tomar clases de baile y
canto</option>
  <option>Programar en HTML</option>
</select>
```

Css para la lista desplegable:

```
select{background: #eee url(arrow.png);
background-position: 280px center;
background-repeat: no-repeat;
padding: 10px; font-size: 16px;
border: 0; border-radius: 3px;
  -webkit-appearance: none;
  -webkit-transition: all 0.4s;
  -moz-transition: all 0.4s;
  transition: all 0.4s;
  -webkit-box-shadow: 0 1px 2px
rgba(0,0,0,0.3); }
select:hover{ background-color: #ddd;}
```

## Botones 3D

Para realizar botones 3d, empezamos creando la estructura base de nuestro documento

html5. Si bien este ejercicio no tiene controles de formulario, esto te puede ayudar a simular un botón genérico.

```
<body>
<a href="#Aqui URL" class="boton colorBoton
formaBoton">PRESIONAME!</a>
</body>
```

Ahora haremos la magia dando los estilos correspondientes a nuestro botón para dale el efecto 3D tal como podés ver en la imagen 1. Recordá que hemos asignado las siguientes clases ( Class=" boton colorRojo formaBoton") estas clases ayudaran a dar el tamaño, color y forma del nuestro botón.

La creación de botones 3D con Css3 ahora son muy usados por todos los front-end porque dan un efecto original visualmente.



1



# Formulario de acceso

Comenzaremos con el código perteneciente al cuerpo del formulario. Mirá también la imagen **1** para que te quede más claro lo que vamos realizando.

```
<div class="container">
  <div class="login">
    <h1>Iniciar Sección</h1>
    <form method="post" action="">
      <p><input type="text" name="login"
value="" placeholder="Usuario o Email"></p>
      <p><input type="password"
name="password" value=""
placeholder="Clave"></p>
      <p class="remember_me">
        <label>
          <label>
            <input type="checkbox" name="remember_
me" id="remember_me">
              Recordarme en este equipo
            </label>
          </label>
        </p>
      <p class="submit">
        <input type="submit" name="commit"
value="Entrar"></p>
```

```
<div class="login-help">
  <p>¿Olvidó su contraseña? <a
href="#">Haga clic aquí para restablecerla</
a></p>
</div>
</form>
</div>
```

Después de terminar el HTML necesitamos comenzar con la parte de CSS del diseño. Lo que haremos es agregar el ancho y alto del formulario y lo colocaremos en el centro de la página.

```
.container { margin: 50px auto; width:
640px;}
.login { position: relative; margin: 0 auto;
padding: 20px 20px 20px; width: 310px;
background: white; border-radius: 3px;
-webkit-box-shadow: 0 0 200px rgba(255,
255, 255, 0.5), 0 1px 2px rgba(0, 0, 0, 0.3);
box-shadow: 0 0 200px rgba(255, 255, 255,
0.5), 0 1px 2px rgba(0, 0, 0, 0.3); }
```

Ahora comenzaremos a darle una forma más agradable. Escribiendo el siguiente código, agregaremos un borde y una sombra gris al formulario, como podés ver en la imagen **2**

## Iniciar Sección



Ahora que hemos agregado estilo a los campos, podríamos querer agregar estilo al botón de enviar, para eso:

```
input[type=submit]:active {background:
#cde5ef;
border-color: #9eb9c2 #b3c0c8 #b4ccce;
-webkit-box-shadow: inset 0 0 3px rgba(0, 0, 0, 0.2);
box-shadow: inset 0 0 3px rgba(0, 0, 0, 0.2);}
```

También agregaremos estilo a la casilla de verificación de la aplicación para recordar la información de acceso del usuario.

Comenzaremos agregando el CSS con el cual daremos diseño a los campos de usuario y contraseña tal como te mostramos en la imagen 1



The image shows a login form titled "Iniciar Sección" (Login Section). It features two input fields: "Usuario o Email" (Username or Email) and "Clave" (Password). Below the password field is a checkbox labeled "Recordarme en este equipo" (Remember me on this device). To the right of the checkbox is a blue button labeled "Entrar" (Enter). At the bottom, there is a link that says "¿Olvidó su contraseña? Haga clic aquí para restablecerla." (Forgot your password? Click here to reset it).

Podemos terminar el ejercicio agregando un color de fondo.

```
body { font: 13px/20px "Lucida Grande",  
Tahoma, Verdana, sans-serif;  
color: #404040; background: #0ca3d2;}
```

Al resultado de todas estas variaciones que venimos aplicando en el aspecto del formulario podés verlo en la figura 1

## Formulario elegante

¡Prestá mucha atención a lo que sigue!  
Ahora te vamos a describir cada uno de los pasos para llevar adelante un formulario elegante como el de la imagen 2.

Este es el código HTML para el cuerpo del formulario:

```
<div id="contact">  
<h1>Formulario de Envio</h1><br><br>  
<form action="#" method="post">  
<fieldset>  
<label for="name">Nombre:</label>  
<input type="text" id="name"
```

```
placeholder="Ingrese su nombre completo"  
><br>  
<label for="email">Email:</label>  
<input type="email" id="email"  
placeholder="ingrese su direccion" />  
<label for="message">Mensaje:</label> <br>  
<textarea id="message" placeholder="Escriba  
aqui..."></textarea><br>  
<input type="submit" value="Enviar" />  
</fieldset>  
</form>  
</div>
```

En un primer momento, se especifica las características de los elementos presentes en la página, atribuyendo el valor 0 a sus propiedades.

```
body, div, h1, form, fieldset, input, textarea {  
margin: 0; padding: 0;  
border: 0; outline: none; }  
html {height: 100%;}
```

El fondo de la página es un degradado de color aplicado al elemento body, con las adaptaciones para los diferentes navegadores:



```
body {
background: #728eaa;
background: -moz-linear-gradient(top,
#25303C 0%, #728EAA
100%);
background: -webkit-gradient(linear, left top,
left bottom,
color-stop(0%,#25303C), color-
stop(100%,#728EAA));
font-family: sans-serif;
}
```

Las etiquetas de los campos label presentan, entre otros estilos, una sombra que detallamos a continuación:

```
label { float: left; clear: left;
margin: 11px 20px 0 0;
width: 95px; text-align: right;
font-size: 16px; color: #445668;
text-transform: uppercase;
text-shadow: 0px 1px 0px #f2f2f2; }
```

Los campos input presentan, en especial, degradados lineales, bordes redondeados, sombras aplicadas a los distintos elementos y al texto.

```
input {
width: 260px; height: 35px;
padding: 5px 20px 0px 20px;
margin: 0 0 20px 0; background: #5E768D;
background: -moz-linear-gradient(top,
#546A7F 0%, #5E768D
20%);
background: -webkit-gradient(linear, left top,
left bottom,
color-stop(0%,#546A7F), color-
stop(20%,#5E768D));
border-radius: 5px; -moz-border-radius: 5px;
-webkit-border-radius: 5px;
-moz-box-shadow: 0px 1px 0px #f2f2f2;
-webkit-box-shadow: 0px 1px 0px #f2f2f2;
font-family: sans-serif; font-size: 16px;
color: #f2f2f2; text-transform: uppercase;
text-shadow: 0px -1px 0px #334f71; }
```

Los valores de ejemplo (placeholder), también presentan estilos de formato, en especial, texto sombreado:

```
input:-webkit-input-placeholder {
color: #a1b2c3; text-shadow: 0px -1px 0px
#38506b; }
input:-moz-placeholder {color: #a1b2c3;
text-shadow: 0px -1px 0px #38506b; }
```

CSS hace más fácil la creación de formularios atractivos rápidamente, porque una vez que se ha desarrollado un diseño, lo podemos reproducir en cada formulario del sitio.



Usando CSS, podemos cambiar muchas cosas sobre el aspecto de nuestros formularios y los elementos dentro de ellos: redefinir la etiqueta de HTML que controla estos elementos, aplicar las llamadas clases css a estas etiquetas, etc.



El campo de texto multilínea textarea presenta las propiedades CSS3 de degradado, bordes redondeados y una sombra bajo el texto.

```
textarea {
width: 260px; height: 170px;
padding: 12px 20px 0px 20px;
margin: 0 0 20px 0; background: #5E768D;
background: -moz-linear-gradient(top,
#546A7F 0%, #5E768D
20%);
background: -webkit-gradient(linear, left top,
left bottom,
color-stop(0%,#546A7F), color-
stop(20%,#5E768D));
border-radius: 5px;
-moz-border-radius: 5px;
-webkit-border-radius: 5px;
-moz-box-shadow: 0px 1px 0px #f2f2f2;
-webkit-box-shadow: 0px 1px 0px #f2f2f2;
font-family: sans-serif; font-size: 16px;
color: #f2f2f2; text-transform: uppercase;
text-shadow: 0px -1px 0px #334f71; }
```

Los ejemplos de valores posibles (placeholder) también presentan estilos de formato y en especial, texto sombreado.

```
textarea::-webkit-input-placeholder {
color: #a1b2c3; text-shadow: 0px -1px 0px
#38506b;
}
textarea:-moz-placeholder {
color: #a1b2c3;
text-shadow: 0px -1px 0px #38506b; }
```

Los campos de inserción de datos (input y textarea) tendrán un formato diferente cuando se estén usando, mediante la pseudo-clase :focus: se les aplicará un degradado.

```
input:focus, textarea:focus {
background: #728eaa;
background: -moz-linear-gradient(top,
#668099 0%, #728eaa
20%);
background: -webkit-gradient(linear, left top,
left bottom,
color-stop(0%,#668099), color-
stop(20%,#728eaa)); }
```

El botón de envío (input[type=submit]) presenta como estilos CSS3 una sombra y un degradado. **(1)**

```
input[type=submit] {
width: 185px; height: 52px;
float: right; padding: 10px 15px;
margin: 20 15px 20 150;
-moz-box-shadow: 0px 0px 5px #999;
-webkit-box-shadow: 0px 0px 5px #999;
border: 1px solid #556f8c;
background: -moz-linear-gradient(top,
#718DA9 0%, #415D79
100%);
background: -webkit-gradient(linear, left
top, left bottom,
color-stop(0%,#718DA9), color-
stop(100%,#415D79));
cursor: pointer; }
h1{color:white; }
```

# Formulario Clásico

Si bien ya vimos algo de formularios, con este ejercicio seguiremos viendo el aspecto estético a través de una amplia definición de estilos. Observá la imagen 1 y seguí con nosotros cada paso. Primero:

```
<form id="login">
<h1>Ingreso</h1>
<fieldset id="inputs">
<input id="username" type="text"
placeholder="Usuario" autofocus required>
<input id="password" type="password"
placeholder="Clave" required>
</fieldset>
<fieldset id="actions">
<input type="submit" id="submit"
value="Ingresar">
<a href="">Olvido su Clave?</a>
<a href="">Registrar</a>
</fieldset>
</form>
```

Al campo de usuario del formulario se le ha aplicado un estilo sombreado mediante seis sombras:

```
#login { position: absolute;
height: 240px; width: 400px; padding: 30px;
-moz-box-shadow:0 0 2px rgba(0, 0, 0, 0.2), 0
1px 1px rgba(0, 0, 0, .2),
0 3px 0 #fff, 0 4px 0 rgba(0, 0, 0, .2), 0 6px 0
#fff, 0 7px 0 rgba(0, 0, 0, .2);
-webkit-box-shadow: 0 0 2px rgba(0, 0, 0,
0.2),0 1px 1px rgba(0, 0, 0, .2),
0 3px 0 #fff, 0 4px 0 rgba(0, 0, 0, .2), 0 6px 0
#fff, 0 7px 0 rgba(0, 0, 0, .2);
box-shadow: 0 0 2px rgba(0, 0, 0, 0.2), 0 1px
1px rgba(0, 0, 0, .2),
0 3px 0 #fff, 0 4px 0 rgba(0, 0, 0, .2), 0 6px 0
#fff, 0 7px 0 rgba(0, 0, 0, .2); }
```

Podemos ver una línea de puntos alrededor del formulario. Ese efecto se ha conseguido con el **pseudo-elemento :before**. En principio, ese pseudo-elemento permite añadir contenido textual delante del elemento al que se aplique:



```
#login:before { content: ''; position: absolute;
z-index: -1;
border: 1px dashed #ccc; top: 5px; bottom:
5px; left: 5px; right: 5px;
-moz-box-shadow: 0 0 0 1px #fff; -webkit-
box-shadow: 0 0 0 1px #fff;
box-shadow: 0 0 0 1px #fff; }
```

Al título del formulario se le ha aplicado una sombra:

```
h1 {
text-shadow: 0 1px 0 rgba(255, 255, 255, .7),
0px 2px 0
rgba(0, 0, 0, .5);
text-transform: uppercase; text-align: center;
color: #666;
margin: 0 0 30px 0; letter-spacing: 4px;
font: normal 26px/1 Verdana, Helvetica;
position: relative; }
```

A cada lado del título tenemos un trazo difuminado:

```
h1:after, h1:before { background-color: #777;
content: ''; height: 1px; position: absolute;
top: 15px; width: 120px; }
h1:after { background-image: -webkit-
```

```
gradient(linear, left top, right
top, from(#777), to(#fff));
background-image: -webkit-linear-
gradient(left, #777, #fff);
background-image: -moz-linear-gradient(left,
#777, #fff);
background-image: -ms-linear-gradient(left,
#777, #fff);
background-image: -o-linear-gradient(left,
#777, #fff);
background-image: linear-gradient(left, #777,
#fff);
right: 0; }
```

```
h1:before { background-image: -webkit-
gradient(linear, right top, left
top, from(#777), to(#fff));
background-image: -webkit-linear-
gradient(right, #777, #fff);
background-image: -moz-linear-gradient(right,
#777, #fff);
background-image: -ms-linear-gradient(right,
#777, #fff);
background-image: -o-linear-gradient(right,
#777, #fff);
background-image: linear-gradient(right, #777,
#fff);
left: 0; }
```



¿Para qué sirve entonces el pseudo-elemento :before?

En principio, ese pseudo-elemento permite añadir contenido textual delante del elemento al que se aplica.





La pseudo-clase `:focus`, se activa cuando el elemento está seleccionado.

Normalmente se aplica a los elementos `<input>` de los formularios cuando están activados y por tanto, se puede escribir directamente en esos campos.

Los campos de inserción de datos se han colocado en un elemento **fieldset**:

```
fieldset {border: 0; padding: 0; margin: 0;}
```

Los campos de inserción `input` presentan un pequeño pictograma en su interior, se trata simplemente de una imagen no repetida que se ha usado para los dos campos. Los campos tienen esquinas redondeadas y una sombra:

```
#inputs input {
background: #f1f1f1 url(login-sprite.png) no-repeat;
padding: 15px 15px 15px 30px; margin: 0 0 10px 0;
width: 353px; border: 1px solid #ccc;
```

```
-moz-border-radius: 5px; -webkit-border-radius: 5px;
border-radius: 5px;
-moz-box-shadow: 0 1px 1px #ccc inset, 0 1px 0 #fff;
-webkit-box-shadow: 0 1px 1px #ccc inset, 0 1px 0 #fff;
box-shadow: 0 1px 1px #ccc inset, 0 1px 0 #fff; }
```

A continuación se ha posicionado la imagen de fondo para cada campo:

```
#username { background-position: 5px -2px !important; }
#password { background-position: 5px -52px !important; }
```

Cuando un campo de inserción de datos esté siendo utilizado, se le aplicará el estilo de **focus**. La **pseudo-clase `:focus`** permite aplicar un diseño específico, en este caso, una sombra:

```
#inputs input:focus { background-color: #fff;
border-color: #e8c291; outline: none;
-moz-box-shadow: 0 0 0 1px #e8c291 inset;
-webkit-box-shadow: 0 0 0 1px #e8c291 inset;
box-shadow: 0 0 0 1px #e8c291 inset; }
```

El botón **Ingresar** sólo sufre desplazamiento vertical.

```
#actions { margin: 25px 0 0 0; }
```

El botón **Ingresar** utiliza el código de identificación **#submit**. Se le ha aplicado principalmente un degradado, un texto sombreado y una sombra de caja:

```
#ffb94b, #fddb6f);
background-image: -ms-linear-gradient(top,
#ffb94b, #fddb6f);
background-image: -o-linear-gradient(top,
#ffb94b, #fddb6f);
background-image: linear-gradient(top,
#ffb94b, #fddb6f); }
#submit:active { outline: none;
-moz-box-shadow: 0 1px 4px rgba(0, 0, 0, 0.5)
inset;
-webkit-box-shadow: 0 1px 4px rgba(0, 0, 0,
0.5) inset;
box-shadow: 0 1px 4px rgba(0, 0, 0, 0.5) inset; }
#submit::-moz-focus-inner { border: none; }
```

El formato de los dos vínculos restantes es muy sencillo:

```
#actions a { color: #3151A2; float: right;
line-height: 35px; margin-left: 10px; }
```

## Formulario interactivo

La estructura de este formulario, como podés observar en la imagen **1**, es bastante clásica, con dos campos, uno para el **Nombre** y otro para la **Contraseña**:

Cuando se use el botón (:focus, :hover y :active), el degradado cambiará:

```
#submit:hover,#submit:focus {
background-color: #fddb6f;
background-image: -webkit-gradient(linear,
left top, left
bottom, from(#ffb94b), to(#fddb6f));
background-image: -webkit-linear-
gradient(top, #ffb94b, #fddb6f);
background-image: -moz-linear-gradient(top,
```



```

<form id="start" action="/">
<h1>Formulario de Inscripcion</h1>
<p>
<label for="name">Nombre</label>
<input id="name" type="text" />
</p>
<p>
<label for="password">Clave</label>
<input id="password" type="password" />
</p>
</form>

```

Las opciones adicionales para el **Código postal**, la **Ciudad** y el **País**, se encuentran en una caja `<div id="options">` de modo que sea posible gestionar la interactividad, justo debajo de los campos anteriores.

En esa misma caja `<div>` se encuentra el vínculo para acceder a las opciones adicionales **Ver más opciones**, en un párrafo `<p>` que utiliza la clase llamada **show**. Esta caja también contiene el vínculo para **Ocultar las opciones**, en un párrafo `<p>` que utiliza la clase llamada **hide**. (1)

El botón de envío (`type="submit"`) ha sido insertado en un párrafo `<p>` en comentario, para que sea posible mostrar el mensaje de confirmación con los estilos CSS. Los dos elementos interactivos, el botón **ENVIAR** y el vínculo de cancelación **cancelar**, han sido insertados en vínculos:

```

<p>
<a class="submit" href="#finish">Enviar</a> o
<a href="#start">Cancelar</a>
</p>

```

Una vez que haya completado y enviado el formulario, aparecerá un mensaje de confirmación. Ese mensaje emerge dentro de una caja `<div id="finish">`:

[Ver más opciones](#)  
[Ocultar las opciones](#)

**Código postal**

**Ciudad**

**País**

Francia



```
<div id="finish">
<p> Inscripcion finalizada! </p>
</div>
```

Comenzamos a definir los estilos, el formulario (form) presenta un diseño con degradados, bordes redondeados y sombras, que se han adaptado para los principales navegadores.

```
form{ background:#f7f7f7;
background:-moz-linear-gradient(90deg, #ccc,
#fff);
background:-webkit-gradient(linear, left top,
left bottom,
from(#fff), to(#ccc));
border:1px solid #aaa; -moz-border-
radius:10px;
-webkit-border-radius:10px; -moz-box-
shadow:0 0 15px #aaa;
-webkit-box-shadow:0 0 15px #aaa;
margin:60px auto 0; padding:20px;
width:440px; }
```

El título del formulario es un h1:

```
h1{ border-bottom:1px solid #ccc; font-
size:11px;
```

```
font-weight:bold; letter-spacing:2px; margin-
bottom:20px;
text-transform:uppercase; }
```

Los párrafos tienen un margen inferior de 0. Observá que se ha usado la **pseudo-clase :last-child** para seleccionar el último párrafo del formulario:

```
form p{ margin-bottom:20px; }
form p:last-child{ margin-bottom:0;}
```

Las etiquetas (label) presentan, entre otras propiedades, un cursor en forma de mano:

```
label{ cursor:pointer; display:block;
float:left; font-size:13px;
font-weight:bold; line-height:28px;
margin-bottom:5px; width:120px; }
```

Las etiquetas cambian de apariencia cuando se pasa el cursor sobre ellas (hover): aparece el carácter » y el texto cambia de color. El selector **form p:hover label** se aplicará en un formulario (form) a un párrafo (p) cuando se señale con el cursor del ratón (:hover), siempre y cuando contenga la etiqueta (label):

Las etiquetas (label)

presentan, entre otras

propiedades, un cursor en

forma de mano y cambian de

apariencia cuando se pasa el

cursor sobre ellas (hover).



```
form p:hover label{ color:#0459b7;}
```

Con la **pseudo-clase :after**, la hoja de estilo generará el contenido (content): » ("»").

```
form p:hover label:after{ content:" <<<"; }
```

Los campos de inserción de texto (input[type=text]) y de contraseña (input[type=password]) presentan un diseño con fondos de color y transparencia, degradados y bordes redondeados:

```
input[type=text], input[type=password]{  
background: rgba(255, 255, 255, 0.9);  
background:-moz-linear-gradient(90deg, #fff,  
#eee);  
background:-webkit-gradient(linear, left top,  
left bottom,  
from(#eee), to(#fff), color-stop(0.2, #fff));  
border:1px solid #aaa; -moz-border-radius:3px;  
-webkit-border-radius:3px; -moz-box-  
shadow:0 0 3px #aaa;  
-webkit-box-shadow:0 0 3px #aaa;  
padding:5px; }
```

Cuando uno de esos campos esté activo, es decir, presente el elemento focus, se modificará su formato con un borde azul y una sombra.

```
input[type=text]:focus,  
input[type=password]:focus{  
border-color:#093c75; -moz-box-shadow:0 0  
3px #0459b7;  
-webkit-box-shadow:0 0 3px #0459b7;  
outline:none; }
```

El menú desplegable para los países (País) mostrará una sombra y un borde cuando se pase el cursor por encima.

```
select{ cursor:pointer; padding:3px; -moz-box-  
shadow:0 0 3px #aaa;  
-webkit-box-shadow:0 0 3px #aaa; }
```

```
select:active, select:focus{ border:1px solid  
#093c75;  
-moz-box-shadow:0 0 3px #0459b7;  
-webkit-box-shadow:0 0 3px #0459b7;  
outline:none; }
```

El botón para enviar el formulario (input[type=submit]) y el vínculo asociado (a.submit) presentan, un fondo degradado, bordes redondeados y una sombra aplicada a la caja y al texto.

\* ! #



¿Qué permite la Pseudo-clase : after?

Permite añadir contenido a un elemento desde la hoja de estilo, al final del elemento.





```
input[type=submit], a.submit{
background:#ddd;
background:-moz-linear-gradient(90deg,
#0459b7, #08adff);
background:-webkit-gradient(linear, left top,
left bottom, from(#08adff), to(#0459b7));
border:1px solid #093c75; -moz-border-
radius:3px;
-webkit-border-radius:3px; -moz-box-
shadow:0 1px 0 #fff;
-webkit-box-shadow:0 1px 0 #fff; color:#fff;
cursor:pointer;
font-family:Arial,sans-serif; font-size:12px;
font-weight:bold; margin-left:120px;
padding:5px 10px;
text-decoration:none; text-shadow:0 1px 1px
#333;
text-transform:uppercase; }
```

Al pasar el cursor sobre ellos (:hover), esos elementos interactivos tendrán un diseño diferente.

```
input[type=submit]:hover, a.submit:hover{
background:#eee;
background:-moz-linear-gradient(90deg,
#067cd3, #0bcdff);
background:-webkit-gradient(linear, left top,
```

```
left bottom,
from(#0bcdff), to(#067cd3));
border-color:#093c75; text-decoration:none; }
```

Por último, esos elementos también mostrarán un diseño específico cuando se haga clic en ellos (pseudo-clase :active) y cuando estén activos (pseudo-clase :focus).

```
input[type=submit]:active,
input[type=submit]:focus,
a.submit:active, a.submit:focus{
background:#ccc;
background:-moz-linear-gradient(90deg,
#0bcdff, #067cd3);
background:-webkit-gradient(linear, left top,
left bottom,
from(#067cd3), to(#0bcdff));
border-color:#093c75; outline:none; }
```

Las opciones del formulario se han insertado en una caja <div id="content">.

```
#options{ margin-bottom:20px; }
```

Al visualizar inicialmente el formulario, podemos ver el vínculo para acceder a las opciones (Ver más opciones).

Recordá que la pseudo-clase :active permite especificar el aspecto de un elemento cuando se hace clic sobre él (y mientras se mantiene el botón del ratón apretado), mientras que la pseudo-clase :focus permite especificar el aspecto de un elemento cuando este tiene el foco.



```
#options.show{ display:block; }
```

Este vínculo usa la clase **.show**.

```
<p class="show"> <a href="#options">Ver más  
opciones</a> </p>
```

Al visualizar inicialmente el formulario, las opciones, que se encuentran dentro de párrafos (<p>), quedan ocultas gracias a la propiedad display con el valor none...

```
#options p{ display:none; }
```

...al igual que los elementos que usan la clase **.hide**.

```
#options .hide{ display:none; }
```

Podés ver todos estos cambios en la imagen 1. Eso es lo que ocurre con el párrafo que contiene el vínculo para ocultar las opciones (Ocultar las opciones).

```
<p class="hide"> <a href="#start">Ocultar las  
opciones</a> </p>
```

El vínculo para acceder a las opciones (Ver más opciones) se encuentra dentro de un párrafo <p> que usa la clase **.show**.

```
<p class="show"> <a href="#options">Ver más  
opciones</a> </p>
```

Cuando el usuario haga clic en el vínculo **Ver más opciones**, se estará haciendo una llamada a los elementos indicados en el href del vínculo: **href="#options"**. Con el uso de

la **pseudo-clase :target**, los párrafos **p** de **#options** serán seleccionados y se mostrarán:

```
#options:target p{ display:block; }
```

Y los elementos de **#options** que tengan la clase **.show** quedarán ocultos, como es el caso del vínculo para acceder a las opciones (Ver más opciones):

```
#options:target .show{ display:none; }
```

Ahora podremos ver el vínculo que permite Ocultar las opciones, así como los campos adicionales. Ahora el vínculo para Ocultar las opciones es visible y se encuentra dentro de un párrafo <p> que utiliza la clase **.hide**.

```
<p class="hide"> <a href="#start">Ocultar las
opciones</a> </p>
```

Cuando el usuario haga clic en ese vínculo, se estará haciendo referencia al valor de **href**, es decir, **#start**, que es el código de identificación del formulario:

```
<form id="start" action="/"> ... </form>
```

De este modo volveremos a visualizar el formulario en su estado inicial.

Cuando completemos el formulario, aparecerá un mensaje de confirmación. Ese mensaje se encuentra dentro de un párrafo `<p>`, en una caja `<div id="finish">`.

```
<div id="finish">
<p> Inscripción finalizada! </p>
</div>
```

inicialmente, esa caja `<div id="finish">` está oculta (`display: none`). Se le ha aplicado un fondo de color con transparencia (`rgba()`) y bordes redondeados.

```
#finish{ background:rgba(65, 166, 42, 0.2);
border:2px solid #41a62a; -moz-border-
```

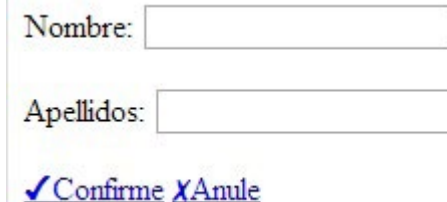
```
radius:3px;
-webkit-border-radius:3px; display:none;
padding:5px 10px; }
```

Cuando el usuario haga clic en el botón de envío, la pseudo-clase `:target` permitirá aplicar esos estilos a la caja `<div id="finish">` y podremos visualizarla.

```
#finish:target { display:block; }
```

## Botones con símbolos

Ahora vamos a crear un formulario clásico con dos botones, uno para validar el formulario y otro para anularlo (imagen 1). Sin embargo, no vamos a usar botones, sino vínculos, ya que la técnica que vamos a utilizar con pseudo-elementos, no se puede aplicar a los botones. Este es el formulario inicial: (2)



```
<form id="registro" method="#"
action="#">
<p> <label for="nombre">Nombre: </
label>
<input type="nombre" name="nombre"
id="nombre" /> </p>
<p> <label for="apellidos">Apellidos: </
label>
<input type="apellidos"
name="apellidos" id="apellidos" /></p>
<p> <a href="#">Confirme</a> <a
href="#">Anule</a> </p>
</form>
```



Nombre:

Apellidos:

[✓ Confirme](#) [X Anule](#)

La técnica de este ejercicio consiste en crear una clase para todos los botones que vaya a usar en su formulario. Esta clase permitirá diseñar el botón con todos los elementos de formato habituales. Podemos usar las propiedades CSS3 porque los vínculos aparecen como bloque en línea: **display: inline-block**. Con esta clase aplicaremos a los botones un ligero degradado y bordes redondeados.

```
.botones { display: inline-block;
border: 1px solid #000; padding: .2em .5em;
margin: 0 4em 0 0;
background: -moz-linear-gradient(top, #fff 0%,
#eee 100%);
background: -webkit-linear-gradient(top, #fff
0%, #eee 100%);
```

```
background: -o-linear-gradient(top, #fff
0%, #ccc 100%);
background: linear-gradient(top, #fff 0%, #ccc
100%);
-moz-border-radius: .3em;
-webkit-border-radius: .3em; border-radius:
.3em;
font-weight: bold; font-size: 1em;
text-decoration: none; color: black; }
```

El pseudo-elemento **::before** permite añadir texto delante del elemento al que esté asociado. El pseudo-elemento **::after** hace lo mismo, pero detrás de dicho elemento. Vamos a aplicar el pseudo-elemento **::before** a la clase genérica de los botones **.botones**, para que el estilo sea más sencillo.

```
.botones:before { font-size: 1em; padding: 0
.5em 0 0; }
```

Ahora tendremos que indicar el contenido que vamos a añadir delante de la etiqueta del botón, el texto del **vínculo a**. Debemos especificar a qué elemento se le añadirá ese contenido, ya que cada botón tendrá un contenido diferente. En nuestro ejemplo (el de la imagen 1), vamos a crear dos selectores específicos **.validar:before** y **.anular:before**, con un estilo específico para cada botón. Segundo parámetro: se trata del pictograma que vamos a insertar. La norma **Unicode UTF-8** nos propone cientos de símbolos y de pictogramas. Para la validación del formulario, queremos insertar el **símbolo de validación**, que tiene el **código de visualización \2714**.



¿Qué es y para qué sirve la norma Unicode UTF-8?

El objetivo de esta norma es terminar con tantos sistemas de codificación incompatibles entre sí de modo que para cualquier carácter del mundo, haya un número que lo identifique unívocamente. UTF-8 es un formato para codificar caracteres Unicode utilizando secuencias de bytes.



Para la anulación del formulario, queremos usar el símbolo de la **cruz de eliminación**, que tiene el **código de visualización \2717**.

Estos son los dos estilos:

```
.validar:before { content: "\2714"; }  
.anular:before { content: "\2717"; }
```

¡Última etapa! ahora vamos a aplicar las clases que acabamos de crear a los **elementos a** de los vínculos del formulario. Los dos vínculos utilizan la clase genérica de los botones **botoncito** y cada uno de ellos utilizará la clase que le corresponda para la inserción de contenido: **validar** y **anular**.

```
<a href="#" class="botoncito  
validar">Confirme</a>  
<a href="#" class="botoncito anular">Anule</a>
```

## Validación de los datos introducidos

Ahora vamos a aprender a modificar el diseño de la validación de los formularios con los estilos CSS3. Vamos a crear un formulario con tres campos y un botón de envío. Este es el código HTML del formulario:

```
<form id="registro" method="#" action="#">  
<p> <label for="nombre">Su nombre: </label>  
<input type="text" id="nombre" required /> </p>  
<p> <label for="email">Su e-mail: </label>
```

```
<input type="email" id="email" required /> </p>  
<p> <label for="edad">Su edad: </label>  
<input type="text" id="edad" pattern="\d{2}" /> </p>  
<p> <input type="submit" id="enviar" value="Enviar" /></p>  
</form>
```

Los dos primeros campos son obligatorios (required) y el último incluye una validación con una expresión regular. Se ha insertado un único botón de envío.

Por otro lado, estos son los estilos que se han usado para aplicar un diseño sencillo a los elementos del formulario:

```
input { border: solid 1px gray;
width: 200px; }
label, input{ display: inline-block; }
label { width: 100px; text-align: right;
margin-right: 10px; }
input[type=submit]{ width: 75px;
background: none; }
```

Como podés observar en la imagen 1, tenemos un campo de tipo email y un campo con un pattern. La validación de los campos se realiza de forma continua, constantemente. Vamos a ver cómo funciona la validación cuando el campo está activo, con el **elemento :focus** y las pseudo-clases **:valid** y **:invalid**. Se ha establecido que aparezca un punto rojo a la derecha del campo cuando el valor no sea válido y un punto verde cuando dicho valor sea válido.

```
input:focus:invalid { background: url(rojo.png)
no-repeat 98% center;}
input:focus:valid {background: url(verde.png)
no-repeat 98% center;}
```

Cuando el usuario hace clic en el campo **Su nombre**, el valor no es válido aún, ya que por el momento no se ha insertado ningún valor. El campo está vacío y, claro, se trata de un campo obligatorio. Por ese motivo aparece el punto rojo. Por último, para evitar que el punto verde aparezca junto al botón de envío cuando se haga clic, utilizamos la pseudo-clase **:not** para excluirlo mediante su código de identificación.

```
input:focus:valid:not(#enviar) {
background: url(verde.png) no-repeat 98%
center; }
```

Vamos ahora a personalizar ese mensaje de error, tal como te mostramos en la imagen 2. En nuestro formulario vamos a añadir el elemento **<span>** inmediatamente después del elemento **<input>** del primer campo **Su nombre y apellidos**. El contenido de **<span>** es el texto que se mostrará cuando el valor introducido no sea válido.

```
<p>
<label for="nombre">Su nombre y apellidos: </
label>
<input type="text" id="nombre" required />
<span class="noValido">Este campo no puede
dejarse vacío</span>
</p>
```

input	:focus	:invalid	+	span.noValido
Para un elemento de formulario de tipo campo de inserción de datos.	Esta pseudo-clase indica que el campo está activo	Esta pseudo-clase indica que el campo no es válido	Es el selector de adyacente directo que seleccionará al hermano inmediato de un elemento. En nuestro ejemplo, seleccionará el elemento inmediatamente después del campo input, es decir, el elemento span.	se aplica a los elementos <span> que utilizan la clase noValido.

Veamos los estilos CSS que le hemos aplicado a la clase **noValido**:

```
span.noValido { display: none;
position: relative; left: 250px;
top: 5px; width: 230px;
border: solid 1px red;
background-color: lightgoldenrodyellow;
padding: 3px; text-align: center; }
```

La propiedad importante es la **visualización**, **display**, que se encuentra en **none**, para no mostrar el mensaje de error. Todo lo demás es la aplicación de las propiedades habituales. Ahora tenemos que hacer que aparezca ese

mensaje si se produce un error, cuando el campo esté activo. Vamos a usar el selector:

```
input:focus:invalid + span.noValido {display:
block; }
```

¡A tener en cuenta! **(1)**

Hemos usado la propiedad display en block para mostrar el mensaje con el diseño que acabamos de establecer. Veamos cómo funciona el mensaje de advertencia.

Este mensaje aparecerá cuando el usuario no haya introducido ningún dato en el campo **Su nombre y apellidos**. En cuanto introduzca un

carácter, el mensaje se ocultará. Seguiremos el mismo principio para el campo de la dirección de e-mail:

```
<p>
<label for="email">Su e-mail: </label>
<input type="email" id="email" required />
<span class="noValido">La dirección de
e-mail no es correcta</span>
</p>
```

Ahora queremos indicarle a los usuarios que el valor que han insertado es válido. Vamos a tomar como ejemplo el campo e-mail. Añadamos un elemento <span> para insertar



el mensaje de validación, como podés ver en la imagen 1.

```
<p> <label for="email">Su e-mail: </label>
<input type="email" id="email" required />
<span class="noValido">La dirección de
e-mail no es correcta</span>
<span class="valido">La dirección de e-mail
es correcta</span> </p>
```

Añadamos los estilos para el formato del mensaje de validación:

```
span.valido { display: none;
position: relative; left: 250px;
top: 5px; width: 230px;
border: solid 1px green;
background-color: lightgreen;
padding: 3px; text-align: center; }
```

Ahora vamos a hacer que aparezca el mensaje de validación tal como figura en la imagen 2. En la sintaxis anterior habíamos utilizado el selector de adyacente directo: +, pero ahora tenemos dos <span> detrás de <input>. Así que esta vez vamos a usar el selector adyacente general ~. El estilo CSS modificado sería:

```
input:focus:invalid ~ span.noValido,
input:focus:valid ~
span.valido {display: block;}
```

Disponemos de otras dos pseudo-clases para validar los números y las fechas. La **pseudo-clase :in-range** se aplica a los valores que se encuentran dentro de los límites definidos y la **pseudo-clase :out-of-range** se aplica a los valores que excedan de esos límites.

Veamos en la imagen 3 una aplicación sencilla con un formulario que contiene un campo de tipo **number**, con un límite inferior de 18 y un límite superior de 30:

```
<form id="test" method="#" action="#">
<p>
<label for="edad">Indique su edad: </label>
<input type="number" min="18" max="30"
step="2" value="25" id="edad" />
</p>
<p> <input type="submit" id="enviar"
value="Enviar" /> </p>
</form>
```

Estos son los estilos CSS: (4)

1

The screenshot shows a web form with the following fields: 'Su nombre y apellidos:' with the value 'Maria', 'Su e-mail:' with the value 'alumno@issd.edu.ar', and 'Su edad:' which is empty. A red error message box on the right says 'La direccion de e-mail no es correcta'. At the bottom is an 'Enviar' button.

2

The screenshot shows the same web form as in image 1, but the email address is now 'alumno@issd.e'. A green success message box on the right says 'La direccion de e-mail es correcta'. The 'Enviar' button is still at the bottom.

3

The screenshot shows a web form with 'Su nombre y apellidos:' (empty), 'Su e-mail:' (empty), and 'Indique su edad:' with a value of 25. The number input field has a green background. At the bottom is an 'Enviar' button.

4

```
<style>
input#edad:out-of-range { background-
color: lightpink; }
input#edad:in-range { background-color:
lightgreen; }
</style>
```

# *Y por si eso fuera poco... ¡Más sobre formularios!*

## Formulario de Compra

Con este formulario en particular veremos el uso de **content** para agregar las flechas a la izquierda de los campos, como así también amplio manejo de espacios y márgenes.

Podés mirar la imagen 1 así te va quedando más claro cada uno de los elementos aplicados.

1

### >>> Datos Personales

Nombre

Email

Tel/Celular

### >>> Direccion Completa

Direccion

Cod. Postal

Ciudad

### >>> Detalle de la Tarjeta

Tipo de Tarjeta

- ☒ VISA ☐ Naranja  
☐ Mastercard

Nro de Tarjeta

Cod Seguridad

Tarjeta

COMPRAR!

The image shows a web form titled "Formulario de Alta" (Registration Form). It is set against a light yellow background. The form is organized into several sections:

- Nombre y apellidos \***: This section is highlighted in yellow. It contains three input fields: "Primer Nombre", "Segundo Nombre", and "Apellido". To the right of these fields, a note reads: "No te olvides de escribir también tu segundo apellido".
- Dirección \***: This section contains four input fields: "Calle, número, piso, Dpto.", "Código postal", "Ciudad", and "Zona". Below the "Código postal" and "Ciudad" fields are two dropdown menus: "Provincia" (with "Cordoba" selected) and "Zona" (with "Centro" selected).
- Email**: A single input field for the email address.
- Teléfono \***: Two input fields for "Fijo" (Fixed) and "Móvil" (Mobile) phone numbers.

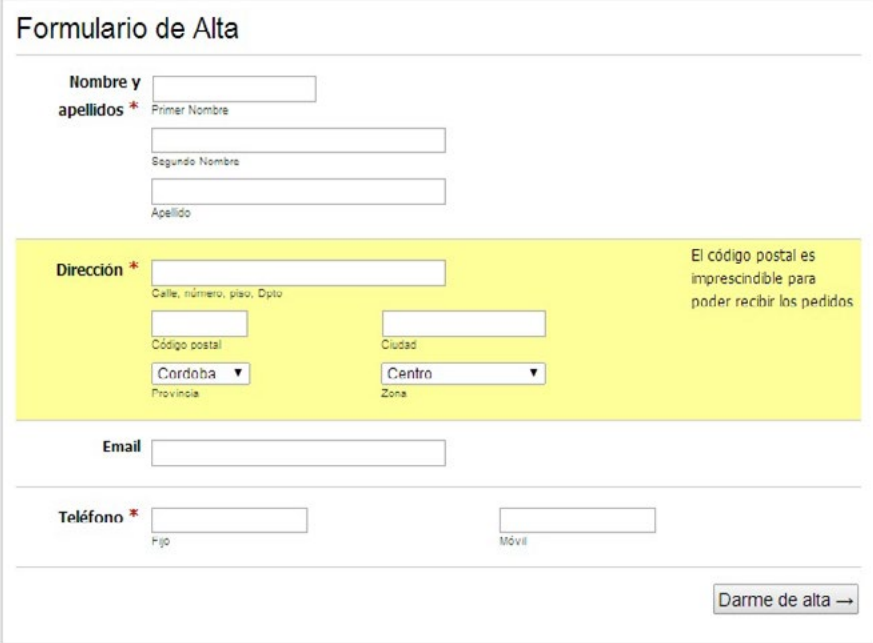
At the bottom right of the form is a button labeled "Dar de alta →".

Detallaremos el estilo para este formulario:

## Formulario de Alta

Este ejercicio fue trabajado en la clase anterior para maquetar solamente los controles en HTML5, y dejarlo a punto con los estilos en esta clase. Ahora vamos a continuarlo para que quede cómo muestra la figura 1.

Desarrollaremos ahora la hoja de estilo (imagen 1) con el siguiente código:



**Formulario de Alta**

**Nombre y apellidos \***

Primer Nombre

Segundo Nombre

Apellido

**Dirección \***

Calle, número, piso, Dpto.

Código postal

Provincia

Ciudad

Zona

El código postal es imprescindible para poder recibir los pedidos

**Email**

**Teléfono \***

Fijo

Móvil

## Formulario de Contacto

En el html tenemos la forma del formulario, dentro de la etiqueta <form> y con una clase para darle un estilo, dentro tenemos una lista desordenada <ul> y cada elemento <li> será un campo para el formulario, esto lo podemos



hacer con <div>, <p> o tablas, pero en este caso utilizaremos los ítems de la lista <li>.

Especificamos los diferentes tipos de <input> para cada campo, por ejemplo <input type="url"> que nos especifica que ese campo es para una dirección de web, y a cada campo lo asociamos con su respectivo <label>. Para definirlo todos los campos son requeridos por que tienen la propiedad **required** y por medio de los **placeholder** de cada campo especificamos el valor esperado de cada campo de entrada. A continuación veremos la estructura para el lenguaje HTML:

```
<form class="contact_form" action="#"
method="post">
  <ul>
    <li> <h2>Contáctanos</h2> </li>
    <li> <label for="name">Nombre:</label>
      <input type="text" placeholder="Johan
Ricardo" required /> </li>
    <li> <label for="email">Email:</label>
      <input type="email" name="email"
placeholder="jricardo@ejemplo.com" required
/> </li>
    <li> <label for="website">Sitio Web:</
label>
```

```
<input type="url" name="web"
placeholder="http://tutosytips.com" required
/> </li>
    <li> <label for="Mensaje">Mensaje:</
label>
      <textarea name="Mensaje" cols="40"
rows="6" required ></textarea>
    </li>
    <li> <button class="submit"
type="submit">Enviar</button> </li>
  </ul>
</form>
```

En lo que sigue detallaremos el código para la hoja de estilo:

```
.contact_form h2 { background: none repeat
scroll 0 0 #F3F3F3;
border-radius: 5px; color: #5CD053;
display: block; font-family: sans-serif;
font-size: 20px; padding: 5px;
text-shadow: 1px 1px 1px #CCCCCC;
width: 433px;}
```

Este código css fue para darle un poco de estilo al título h2 que está dentro del formulario y que podés ver en la imagen 1.

The image shows a web form titled "Contactanos" in green text. The form has a light gray background and a white border. It contains four input fields: "Nombre:" with the value "Ricardo", "Email:", "Sitio Web:" with the value "http://www.ejemplo.com", and "Mensaje:" which is a large text area. At the bottom right of the form is a button labeled "Enviar".

```
.contact_form ul { width:750px;
list-style-type:none; margin:0px;
padding:0px; }

.contact_form li{ padding:12px;
border-bottom:1px solid #eee;
position:relative; }
```

Aplicamos los estilos para los <li> y <ul> del formulario que son los contenedores de cada campo con su label, damos una anchura a la lista y quitamos los margin y padding y de igual modo posicionamos los ítems <li> y colocamos bordes debajo de cada <li>.

```
contact_form label { color: #555555; display: inline-block; float: left; font-family: sans-serif; font-size: 13px; font-weight: bold; margin-top: 3px; padding: 3px; width: 90px; }
```

Formatearemos los controles del formulario:

```
.contact_form input { height:20px; width:220px; padding:5px 8px; }
.contact_form textarea { padding:8px; width:300px; }
.contact_form button { margin-left:90px; }
```

Con el anterior código damos estilos a los <input>, damos un ancho, alto y padding, de igual manera al textarea y al botón de enviar le damos un margin left para posicionarlo.

```
.contact_form input, .contact_form textarea { border:1px solid #aaa;
```

```
box-shadow: 0px 0px 3px #ccc, 0 10px 15px #eee inset;
```

```
border-radius:2px; color: #888; font-size: 12px;
```

```
padding-right:30px; -moz-transition: padding .25s;
```

```
-webkit-transition: padding .25s;
-o-transition: padding .25s;
transition: padding .25s;}
```

```
.contact_form input:focus, .contact_form
textarea:focus {
background: #fff; border:1px solid #555;
box-shadow: 0 0 3px #aaa; padding-
right:80px;}
```

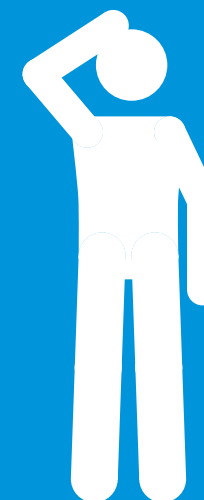
Con los estilos anteriores damos un aspecto visual agradable a los campos de entrada de texto, entre ellos un borde, una sombra interna por medio de box-shadow inset, bordes redondeados, padding y una transición que se hace con el padding que simula una animación que cambia del estado normal a focus, en este estado cambiamos el background, el color del borde y el box-shadow.

<Formulario de Alta>

<Formulario de Compra>

<Formulario de Contacto>

<Formulario con Validación>



```
.contact_form input:required:valid, .contact_
form tex-tarea:required:valid {
    box-shadow: 0 0 5px #5cd053; border-
color: #28921f;}
```

```
.contact_form input:focus:invalid, .contact_
form textarea:focus:invalid {
    box-shadow: 0 0 5px #d45252; border-
color: #b03535}
```

Con el código anterior lo que hacemos es darle las propiedades a los diferentes estados de los campos de entrada, el primero es verde y se coloca cuando hay información y es válida, esto va de acuerdo al tipo de entrada del campo, y se coloca rojo cuando está en estado focus o la entrada es invalida.

```
button.submit { padding: 9px 17px;
    font-family: Helvetica, Arial, sans-serif;
    font-weight: bold; line-height: 1; color: #444;
    border: none; text-shadow: 0 1px 1px
    rgba(255, 255, 255, 0.85);
    background-image: -webkit-gradient(linear,
    0% 0%, 0% 100%, from(#fff), to(#bbb));
    background-image: -moz-linear-gradient(0%
    100% 90deg, #BBBBBB, #FFFFFFF);
```

```
background-color: #fff; border: 1px solid
#f1f1f1;
    border-radius: 10px; box-shadow: 0 1px 2px
    rgba(0, 0, 0, 0.5);}
button.submit:hover { opacity:.55; cursor:
    pointer; color: #fff; }
button.submit:active { border: 1px solid
    #222;
    box-shadow: 0 0 10px 5px #444 inset; }
```

Y por ultimo, como te mostramos en la imagen 1, damos los estilos al botón de envío del formulario: damos un degradado a ese botón, sombras en el botón y en el texto, bordes redondeado entre otros. También damos los estilos en el estado hover del botón que es cuando el mouse esta sobre él y el estado active que se ejecuta al momento de clicar el botón.

The image shows a contact form titled "Contactanos" in green text. It has four input fields: "Nombre:" with the value "Ricardo", "Email:", "Sitio Web:" with the value "http://www.ejemplo.com", and "Mensaje:". Below the fields is a button labeled "Enviar". The form is styled with a light gray background and rounded corners. The "Enviar" button has a gradient background and a shadow.

## Formulario con validación

Con este ejercicio se podrá ver como actua la propiedad required y a su vez seguiremos ampliando la hoja de estilos. A continuación veremos la estructura del formulario. También podes ver su aspecto en la imagen 1.

El estilo tiene la siguiente codificacion:

1

### Reclamos de Area

*Complete todos los campos del formulario, su reclamo tendra respuesta a la brevedad. Gracias!*

Nombre: \*

Email: \*

Tel Celular:

Area:

Soporte Tecnico

▼

Mensaje: \*

Su mensaje no debe superar los 30 caracteres

\* Campo Obligatorio

! Completa este campo

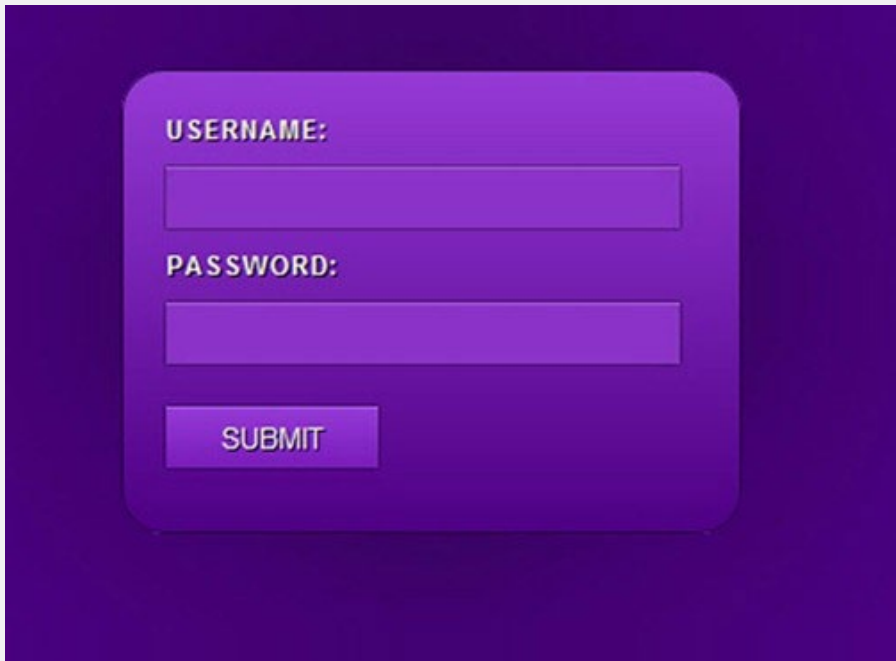




## *Desempeños*

### Desempeño 1

Programá y diseñá los siguientes formularios, aplicá Css y eventos donde se puedan dar. ¡Buena suerte!



**USERNAME:**

**PASSWORD:**

**SUBMIT**

→ Desempeño 2

→ Desempeño 3

Name

E-mail

Website

Details

Submit

## FORMULARIO DE CONTACTO

NOMBRE:

EMAIL:

MENSAJE:

ENVIAR COMENTARIO



→ Desempeño 4

→ Desempeño 5

41

ISSI

## CONTACTO

Visitenos en  
Colón 165, Piso 7, Depto. A  
(M5500GEB) Mendoza, Argentina

Teléfono  
(+54 261) 524 6778

Correo electrónico  
in@envero.org

NOMBRE

CORREO ELECTRÓNICO

TELÉFONO

MENSAJE / CONSULTA

ENVIAR

### Formulario Contacto

Nombre (requerido)

Email (requerido)

Asunto

Mensaje (requerido)

Enviar Mensaje



## Autoevaluación

- 1 | Investiga cómo se le asignan eventos a los botones.
- 2 | Explica con tus palabras lo que realizan las siguientes sentencias:

1

```
.contact_form input:required:valid,  
.contact_form  
textarea:required:valid {  
    background: #fff  
    url(images/valid.png) no-repeat  
    98% center;  
    box-shadow: 0 0 5px  
    #5cd053; border-color: #28921f; }
```

2

```
button.submit:hover { opacity: .85;  
cursor: pointer; }
```

3

```
button.submit:active { border: 1px  
solid #20911e; box-shadow: 0 0 10px  
5px #356b0b inset;  
    -webkit-box-shadow: 0 0 10px 5px  
    #356b0b inset;  
    -moz-box-shadow: 0 0 10px 5px  
    #356b0b inset;  
    -ms-box-shadow: 0 0 10px 5px  
    #356b0b inset;  
    -o-box-shadow: 0 0 10px 5px  
    #356b0b inset; }
```

4

```
.contact_form input:focus, .contact_  
form textarea:focus {background: #fff;  
border:1px solid #555;  
box-shadow: 0 0 3px #aaa; padding-  
right:80px;}
```

5

```
input[type="submit"]:hover,  
input[type="submit"]:active {cursor:  
pointer; background: #fff;  
color: #593131; }
```





## Autoevaluación

43

6

```
#failure, #success { color:
#6EA070; display:none; }
```

7

```
input::-webkit-input-placeholder {
color: #ddd;
text-shadow: 0px -1px
0px #38506b; text-
transform:capitalize; }
```

8

```
.cuadro input:checked + article {
transition: height 0.7s ease-in-out,
box-shadow 0.3s linear;
box-shadow: 0px 0px 0px 1px
rgba(255, 255, 255, 0.8); }
```


3 | Explicá y ejemplificá el atributo READONLY.

4 | Explicá y ejemplificá el atributo FORMTARGET.

5 | ¿Qué implica validar un formulario en forma nativa?

6 | Realizá un cuadro con los distintos atributos de la etiqueta INPUT





¡Llegamos al final de la clase! Con estos ocho encuentros ya debés haber incorporado las funciones principales de HTML y CSS, dos pilares fundamentales para nuestra disciplina. El próximo tema que abordaremos será JavaScript, fundamental para mejorar el aspecto y el control del navegador y de algunos eventos. Esperamos que nos sigas acompañando como hasta ahora.

*¡Hasta la próxima!*