

DESARROLLO DE SISTEMAS WEB

CSS visual y modelo de caja

Colores, tipografías, box model, márgenes, bordes y espaciado. Clase práctica con código y resultado visual.

CLASE 5

06/05/2026

VIRTUAL



CONTEXTO

Dónde estamos en la cursada

- Ya construimos estructura HTML: títulos, párrafos, listas, enlaces, imágenes, tablas y formularios.
- Ya empezamos CSS con reglas, selectores, cascada, herencia y especificidad.
- Hoy pasamos a controlar la apariencia real de una interfaz: color, texto y espacio.
- La idea no es decorar: es hacer que la página se lea, se entienda y se use mejor.

OBJETIVOS

Qué deberían poder hacer al final

Estilo visual

Aplicar colores, fondos, tipografías, tamaños y propiedades de texto.

Espaciado

Diferenciar padding y margin para ordenar elementos en pantalla.

Componentes

Construir una tarjeta simple con HTML y CSS, viendo código y resultado.

REPASO

Una regla CSS siempre dice “a quién” y “qué cambio”

El selector indica qué elemento se modifica. La propiedad y el valor indican qué estilo se aplica.

CÓDIGO

```
p {
  color: blue;
}
```

CSS

CÓMO SE VE

Este párrafo queda azul porque el selector apunta a p.

ANTES Y DESPUÉS

El mismo HTML puede verse muy distinto

HTML define la estructura. CSS define cómo se presenta esa estructura.

SIN CSS PENSADO

Curso de Front-End

Aprendé HTML, CSS y JavaScript desde cero.

[Inscribirme](#)

CON CSS APLICADO

Curso de Front-End

Aprendé HTML, CSS y JavaScript desde cero.

Inscribirme

COLORES

Tres lugares típicos donde aplicamos color

Texto, fondo y borde son los primeros puntos donde CSS empieza a construir identidad visual.

CÓDIGO

```
.aviso {
  color: #1e3a8a;
  background-color: #eef2ff;
  border: 2px solid #818cf8;
}
```

CSS

CÓMO SE VE

Texto azul, fondo suave y borde visible.

COLORES

Colores por nombre: sirven para aprender, pero son limitados

El navegador entiende nombres como red, blue o gray. Para diseño real conviene usar valores más precisos.

CÓDIGO

```
h2 {
  color: blue;
}

p {
  color: gray;
}
```

CSS

CÓMO SE VE

Título azul

Párrafo gris con color por nombre.

Útil al inicio, poco preciso para proyectos

COLORES

HEX: una forma precisa y muy usada

HEX permite trabajar con colores específicos, consistentes y fáciles de reutilizar.

CÓDIGO

CSS

```
.titulo {
  color: #1e40af;
}

.caja {
  background-color: #f8fafc;
}
```

CÓMO SE VE

Color HEX

Fondo claro y título azul controlado con códigos.

COLORES

RGBA permite transparencia

El cuarto valor indica opacidad. 1 es totalmente opaco, 0 es transparente.

CÓDIGO

```
.capa {
  background-color: rgba(15, 23, 42, 0.72);
  color: white;
}
```

CSS

CÓMO SE VE



Ideal para capas sobre fondos

LEGIBILIDAD

Color lindo no siempre significa color legible

El contraste es una decisión funcional. Si el usuario tiene que esforzarse, el diseño está fallando.

MAL CONTRASTE

Texto gris claro sobre fondo blanco. Cuesta leerlo.

```
p {
  color: #cccccc;
  background: #ffffff;
}
```

CSS

MEJOR CONTRASTE

Texto oscuro sobre fondo claro. Se lee sin esfuerzo.

```
p {
  color: #333333;
  background: #ffffff;
}
```

CSS

JERARQUÍA

No todos los textos deberían tener el mismo peso visual

El color ayuda a separar título, descripción, dato secundario y alerta.

CÓDIGO

```
.card h2 { color: #111827; }  
.card p { color: #4b5563; }  
.card .dato { color: #2563eb; }  
.card .alerta { color: #dc2626; }
```

CSS

CÓMO SE VE

Turno confirmado

La consulta quedó agendada correctamente.

Miércoles 18:00 hs

Recordar llevar DNI

TIPOGRAFÍAS

font-family define la familia tipográfica

Se suelen escribir varias opciones. Si la primera no está disponible, el navegador usa la siguiente.

CÓDIGO

```
body {  
  font-family: Arial, sans-serif;  
}
```

CSS

CÓMO SE VE

Texto con Arial

Una sans-serif simple, clara y habitual en interfaces.

Siempre conviene dejar una familia genérica al final

TIPOGRAFÍAS

Tres familias que tienen que reconocer

No hace falta ser diseñador para tomar una decisión razonable de lectura.

Serif

Con remates. Frecuente en textos editoriales.

```
font-family: serif;
```

Sans-serif

Sin remates. Muy usada en interfaces web.

```
font-family: sans-serif;
```

Monospace

Todas las letras ocupan el mismo ancho.
Útil para código.

```
font-family: monospace;
```

Para interfaces, normalmente sans-serif

TIPOGRAFÍAS

font-size crea jerarquía

Título, subtítulo y párrafo no deberían competir con el mismo tamaño.

CÓDIGO

```
h1 { font-size: 36px; }  
h2 { font-size: 24px; }  
p  { font-size: 16px; }
```

CSS

CÓMO SE VE

Título principal

Subtítulo

Texto de lectura normal.

El tamaño comunica importancia

UNIDADES

px es directo; rem escala mejor

Para empezar, px es fácil de entender. rem ayuda a mantener proporciones más consistentes.

CÓDIGO

```
body { font-size: 16px; }

h1 { font-size: 2rem; } /* 32px */
p   { font-size: 1rem; } /* 16px */
```

CSS

CÓMO SE VE

2rem

1rem mantiene una relación clara con el tamaño base.

TIPOGRAFÍAS

font-weight controla el grosor

La negrita sirve para jerarquizar. Si todo está en negrita, nada destaca.

CÓDIGO

```
.normal { font-weight: 400; }  
.fuerte { font-weight: 700; }  
.liviano { font-weight: 300; }
```

CSS

CÓMO SE VE

Texto liviano

Texto normal

Texto fuerte

TIPOGRAFÍAS

font-style permite usar itálica

Sirve para énfasis, citas o pequeñas aclaraciones. No conviene usarlo en bloques largos.

CÓDIGO

```
.frase {
  font-style: italic;
}
```

CSS

CÓMO SE VE

“La itálica funciona bien para una frase breve, no para todo un texto largo.”

Énfasis puntual

LECTURA

line-height mejora muchísimo la lectura

Controla el espacio vertical entre líneas. En textos largos es una propiedad clave.

CÓDIGO

```
p {  
  line-height: 1.6;  
}
```

CSS

CÓMO SE VE

Este párrafo tiene más aire entre líneas. El contenido se vuelve más cómodo de leer y menos apretado.

Más aire, mejor lectura

TEXTO

text-align no se usa igual en todos los casos

Centrar puede funcionar en títulos o textos cortos. En párrafos largos suele ser peor para leer.

CÓDIGO

```
h1 { text-align: center; }  
p  { text-align: left; }
```

CSS

CÓMO SE VE

Título centrado

El párrafo queda alineado a la izquierda para que la lectura sea más natural.

Centrar todo suele ser mala decisión

ENLACES

text-decoration modifica el subrayado

Si sacamos el subrayado, el enlace tiene que seguir pareciendo una acción clickeable.

CÓDIGO

```
a {
  color: #2563eb;
  text-decoration: none;
  font-weight: 700;
}
```

CSS

CÓMO SE VE

Podés [ver el detalle del curso](#) desde este enlace.

No ocultar la intención del enlace

TEXTO

text-transform y letter-spacing sirven para etiquetas cortas

Funcionan bien en categorías, chips o rótulos. Mal usadas, dificultan la lectura.

CÓDIGO

```
.etiqueta {  
  text-transform: uppercase;  
  letter-spacing: 2px;  
  font-size: 12px;  
}
```

CSS

CÓMO SE VE

CLASE VIRTUAL

Usar en textos breves

ESPACIO

Muchas páginas no fallan por el color: fallan por el espacio

APRETADO

Inscripción

Completá tus datos para participar del curso.

Enviar

CON AIRE

Inscripción

Completá tus datos para participar del curso.

Enviar

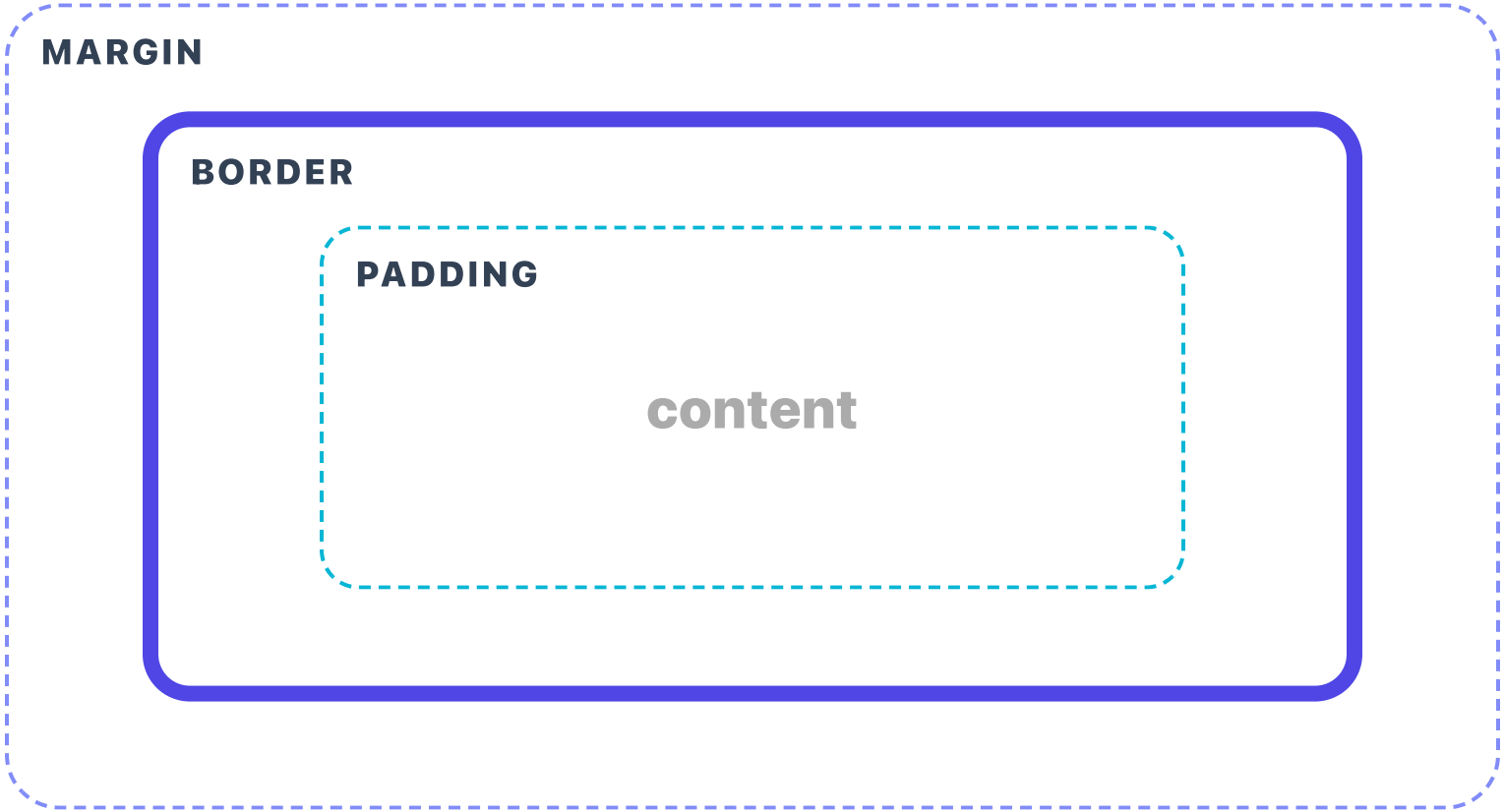
El espaciado ayuda a separar ideas y a que la interfaz no parezca improvisada.

El espacio también comunica

BOX MODEL

En CSS, cada elemento es una caja

Un título, un párrafo, un botón, una imagen o un div ocupan una caja en la pantalla.



BOX MODEL

Content es lo que vive dentro del elemento

Puede ser texto, una imagen, un enlace, un botón o cualquier contenido HTML.

CÓDIGO

HTML

```
<div class="contenido-demo">
  <p>Este texto es el contenido del párrafo.</p>
  <button>Enviar</button>
</div>
```

CSS

```
.contenido-demo {
  font-size: 22px;
  display: grid;
  gap: 12px;
  place-items: center;
}

.contenido-demo button {
  background: #4f46e5;
  color: white;
  border: 0;
  border-radius: 10px;
  padding: 8px 16px;
}
```

CÓMO SE VE



La caja empieza por su contenido

TAMAÑO

width y height controlan dimensiones

El ancho suele definirse más que el alto. Las alturas fijas pueden traer problemas si el contenido crece.

CÓDIGO

```
.caja {  
  width: 300px;  
  height: 140px;  
}
```

CSS

CÓMO SE VE



ESPACIO INTERNO

padding separa el contenido del borde

Si el texto está pegado al borde, falta padding.

CÓDIGO

```
.card {  
  padding: 24px;  
  border: 1px solid #ddd;  
}
```

CSS

CÓMO SE VE

El texto respira porque tiene espacio interno.

Padding = hacia adentro

ESPACIO INTERNO

padding puede escribirse resumido

Con dos valores: el primero aplica arriba/abajo y el segundo izquierda/derecha.

CÓDIGO

```
.boton {  
  padding: 12px 24px;  
}
```

CSS

CÓMO SE VE



12px vertical · 24px horizontal

ESPACIO EXTERNO

margin separa una caja de otras cajas

Si los elementos están pegados entre sí, probablemente falta margin.

CÓDIGO

```
.bloque {  
  margin-bottom: 24px;  
}
```

CSS

CÓMO SE VE



Margin = hacia afuera

ESPACIO EXTERNO

margin también puede aplicarse por lado

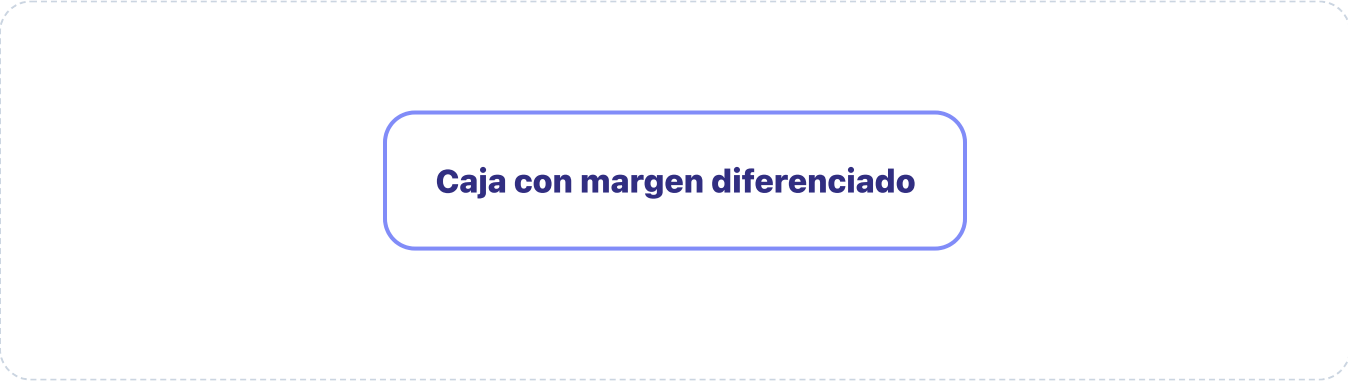
El orden completo es arriba, derecha, abajo, izquierda.

CÓDIGO

```
.box {  
  margin: 20px 10px 30px 10px;  
}
```

CSS

CÓMO SE VE



Caja con margen diferenciado

CENTRADO

margin: 0 auto centra bloques con ancho definido

Funciona cuando el elemento tiene un width definido.

CÓDIGO

```
.card {  
  width: 360px;  
  margin: 0 auto;  
}
```

CSS

CÓMO SE VE

Caja centrada con margin auto

DIFERENCIA CLAVE

Padding y margin no son lo mismo

PADDING

CSS

```
.box {  
  padding: 24px;  
}
```

Texto separado del borde

MARGIN

CSS

```
.box {  
  margin: 24px;  
}
```

Caja separada de afuera

Pregunta rápida: ¿quiero separar el contenido del borde o quiero separar este elemento de otros?

Esta diferencia evita muchos errores

BORDES

border define grosor, estilo y color

Un borde puede separar visualmente un componente sin hacerlo pesado.

CÓDIGO

```
.card {  
  border: 1px solid #d1d5db;  
}
```

CSS

CÓMO SE VE



BORDES

El estilo del borde también cambia el mensaje visual

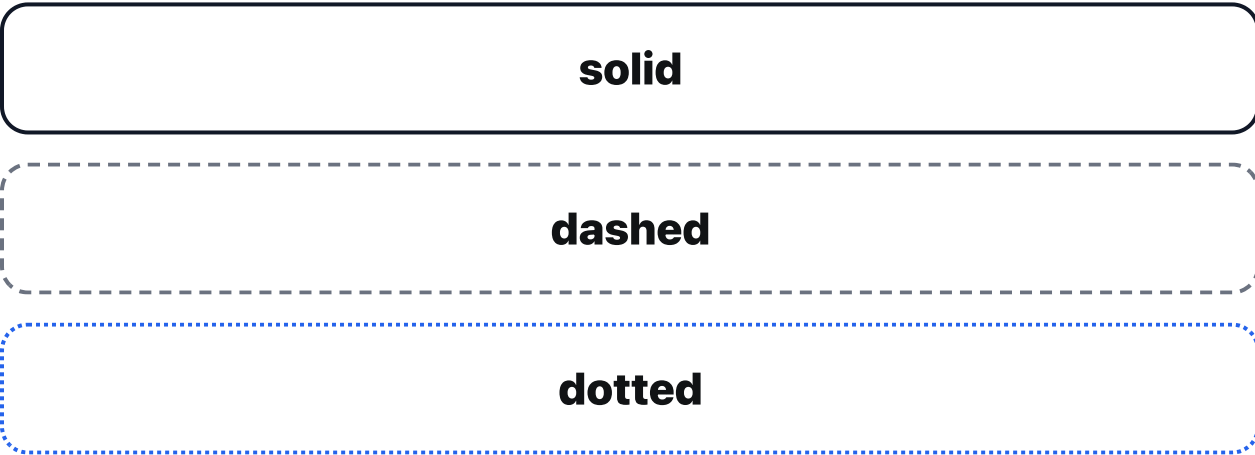
solid es el más común, pero dashed y dotted sirven para estados o zonas de carga.

CÓDIGO

```
.a { border: 2px solid #111827; }  
.b { border: 2px dashed #6b7280; }  
.c { border: 2px dotted #2563eb; }
```

CSS

CÓMO SE VE



BORDES

border-left sirve para destacados y alertas

Un borde lateral comunica énfasis sin llenar toda la caja de color fuerte.

CÓDIGO

```
.alerta {  
  border-left: 5px solid #dc2626;  
  background: #fef2f2;  
  padding: 16px;  
}
```

CSS

CÓMO SE VE

Atención: falta completar un campo obligatorio.

BORDES

border-radius redondea esquinas

El redondeo suele hacer que cards, botones e inputs se vean más actuales.

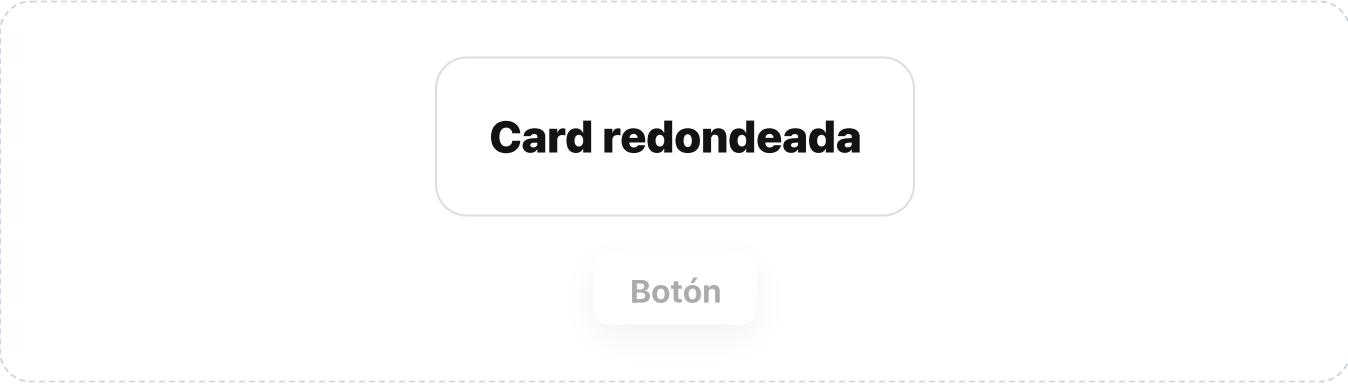
CÓDIGO

```
.card {
  border-radius: 16px;
}

button {
  border-radius: 8px;
}
```

CSS

CÓMO SE VE



No abusar: consistencia primero

BORDES

border-radius: 50% puede crear círculos

Si el elemento tiene el mismo width y height, 50% genera un círculo.

CÓDIGO

```
.avatar {  
  width: 96px;  
  height: 96px;  
  border-radius: 50%;  
}
```

CSS

CÓMO SE VE

EM

Útil para avatares o iniciales

TAMAÑO REAL

El ancho real puede sorprender

Sin border-box, al width se le suma el padding y el borde.

CONTENT-BOX

CSS

```
.box {
  width: 260px;
  padding: 24px;
  border: 4px solid #818cf8;
}
```

width 260 + padding + border

BORDER-BOX

CSS

```
.box {
  box-sizing: border-box;
  width: 260px;
  padding: 24px;
  border: 4px solid #818cf8;
}
```

todo entra en 260px

box-sizing vuelve el layout más predecible

BASE CSS

Un reset mínimo evita comportamientos molestos

No reemplaza aprender CSS, pero ayuda a empezar con una base más controlada.

CÓDIGO

```
* {
  box-sizing: border-box;
}

body {
  margin: 0;
  font-family: Arial, sans-serif;
  background: #f5f5f5;
}
```

CSS

CÓMO SE VE

Base limpia

Sin margen inesperado del body y con box-sizing predecible.

COMPONENTE

HTML de una tarjeta simple

La estructura HTML tiene que ser clara antes de pensar en el diseño.

CÓDIGO

```
<div class="card">
  <h2>Curso de Front-End</h2>
  <p>Aprendé HTML, CSS y JavaScript desde cero.</p>
  <a href="#">Ver más</a>
</div>
```

HTML

CÓMO SE VE

Curso de Front-End

Aprendé HTML, CSS y JavaScript desde cero.

[Ver más](#)

COMPONENTE

CSS base de la tarjeta

Acá aparecen juntos width, margin, padding, fondo, borde y border-radius.

CÓDIGO

```
.card {
  width: 340px;
  margin: 40px auto;
  padding: 24px;
  background: white;
  border: 1px solid #ddd;
  border-radius: 16px;
}
```

CSS

CÓMO SE VE



La tarjeta ya tiene forma visual

COMPONENTE

Mejoramos los textos de la tarjeta

El diseño mejora cuando cada elemento tiene una jerarquía clara.

CÓDIGO

```
.card h2 {  
  margin-top: 0;  
  color: #1e3a8a;  
  font-size: 24px;  
}  
  
.card p {  
  color: #4b5563;  
  line-height: 1.5;  
}
```

CSS

CÓMO SE VE



Título, texto y acción no pesan igual

COMPONENTE

El enlace puede verse como botón

El HTML sigue siendo un enlace, pero visualmente comunica una acción.

CÓDIGO

```
.card a {
  display: inline-block;
  margin-top: 12px;
  padding: 10px 16px;
  background: #1e3a8a;
  color: white;
  border-radius: 8px;
  text-decoration: none;
}
```

CSS

CÓMO SE VE



display: inline-block ayuda al padding

DETALLE IMPORTANTE

¿Por qué display: inline-block?

Un enlace es inline por defecto. inline-block permite que se comporte mejor como botón visual.

CÓDIGO

```
a {
  display: inline-block;
  padding: 10px 16px;
  margin-top: 12px;
}
```

CSS

CÓMO SE VE

Texto antes del enlace.

Acción con forma de botón

Pequeña propiedad, gran diferencia

EJEMPLO COMPLETO

Código completo de la tarjeta

HTML

```
<div class="card">
  <h2>Curso de Front-End</h2>
  <p>Aprendé HTML, CSS y JavaScript desde cero.</p>
  <a href="#">Ver más</a>
</div>
```

HTML

CSS

```
.card {
  width: 340px;
  margin: 40px auto;
  padding: 24px;
  background: white;
  border: 1px solid #ddd;
  border-radius: 16px;
}

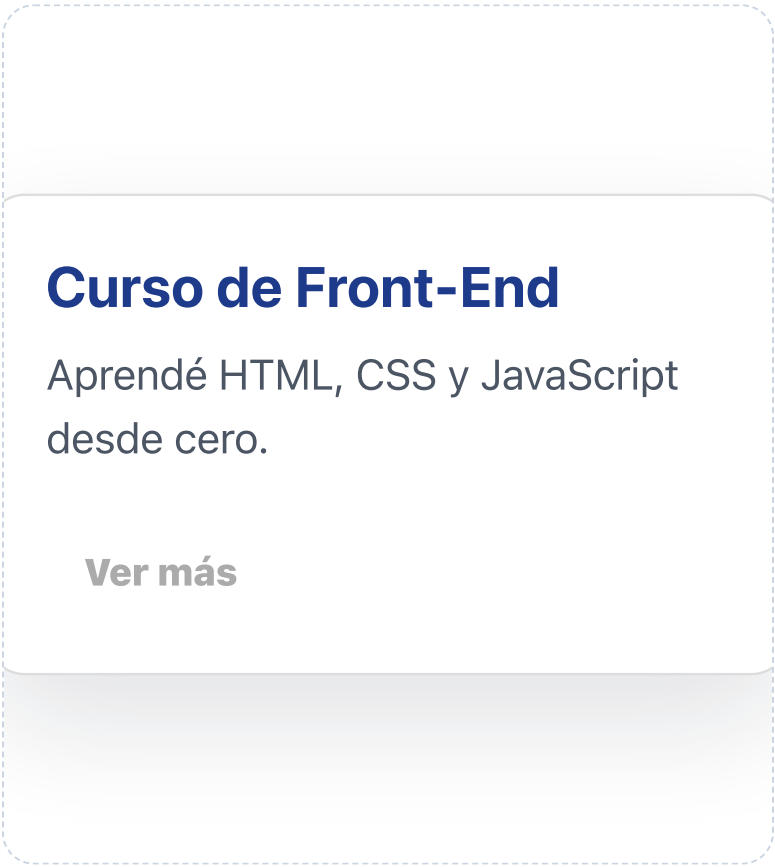
.card a {
  display: inline-block;
  padding: 10px 16px;
  background: #1e3a8a;
  color: white;
  border-radius: 8px;
  text-decoration: none;
}
```

CSS

RESULTADO

Así debería verse el componente terminado

No es “diseño avanzado”, pero ya hay estructura, jerarquía, color, aire y acción.



ERRORES COMUNES

Qué revisar antes de entregar

Problemas frecuentes

- Colores con poco contraste.
- Demasiados colores distintos.
- Todo centrado sin criterio.
- Textos pegados al borde.
- Confundir padding y margin.

Buenas decisiones

- Paleta simple.
- Jerarquía tipográfica clara.
- Espaciado consistente.
- Bordes suaves.
- box-sizing: border-box desde el inicio.

PRÁCTICA GUIADA

Transformar HTML crudo en una presentación visual

Consigna en clase

Partimos de un bloque HTML simple y lo convertimos en una tarjeta de presentación.

1. Definir una clase para la sección principal.
2. Aplicar width, margin y padding.
3. Agregar color de fondo, borde y border-radius.
4. Mejorar título, párrafo y enlace.

PRÁCTICA GUIADA

HTML de partida

CÓDIGO

```
<section class="presentacion">
  <h1>Hola, soy Ana</h1>
  <p>Estoy aprendiendo desarrollo web front-end.</p>
  <a href="#">Contactarme</a>
</section>
```

HTML

CÓMO SE VE SIN CSS

Hola, soy Ana

Estoy aprendiendo desarrollo web front-end.

Contactarme

PRÁCTICA GUIADA

CSS posible para la presentación

No es la única solución. Lo importante es justificar cada propiedad usada.

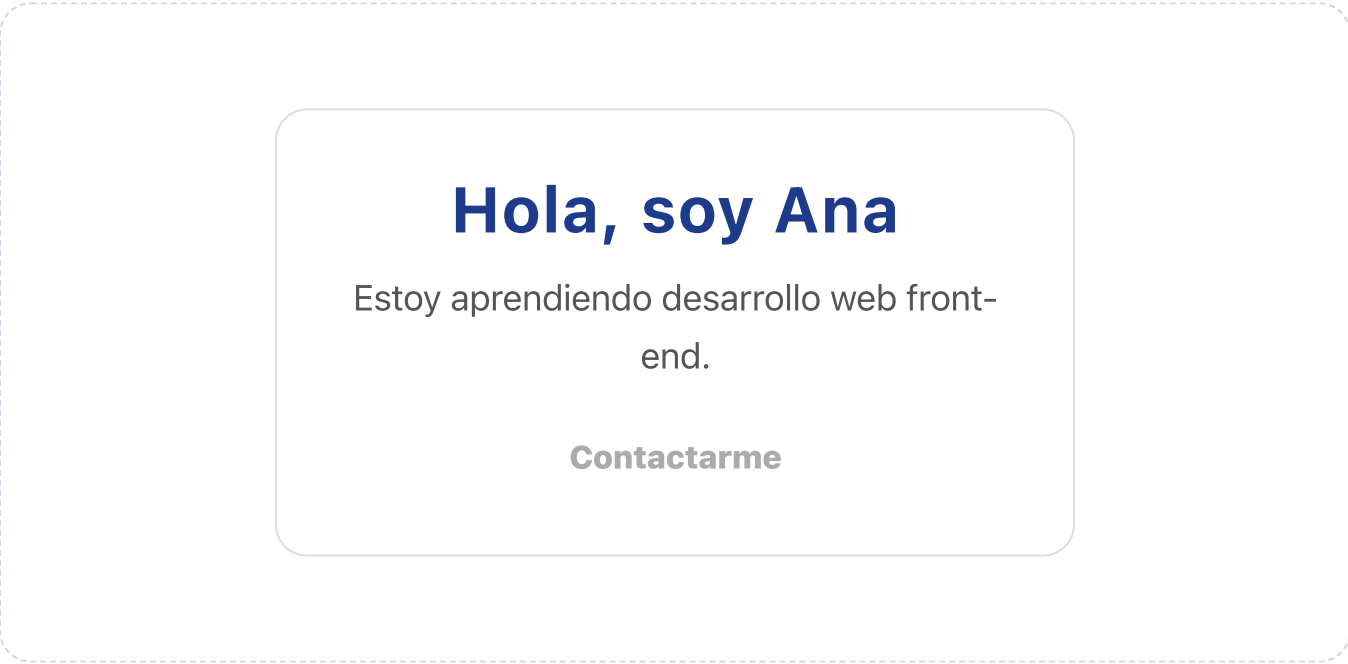
CÓDIGO

```
.presentacion {
  width: 400px;
  margin: 50px auto;
  padding: 30px;
  background: white;
  border: 1px solid #ddd;
  border-radius: 16px;
  text-align: center;
}

.presentacion p {
  color: #555;
  line-height: 1.6;
}
```

CSS

CÓMO SE VE



Una solución posible

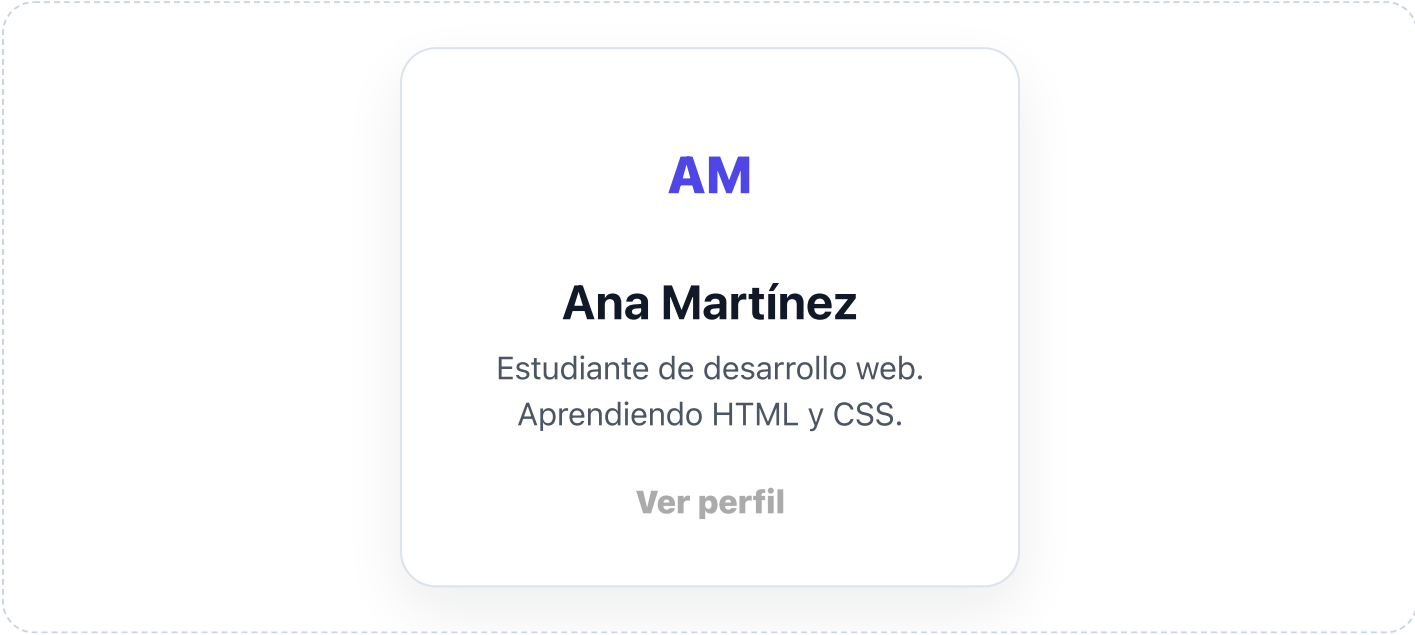
EJERCICIO INTEGRADOR

Crear una tarjeta de presentación personal

Debe incluir

- Nombre o título principal.
- Descripción breve.
- Imagen, iniciales o ícono simple.
- Botón o enlace.
- Colores, tipografía, padding, margin, borde y border-radius.

RESULTADO ESPERADO ORIENTATIVO



ENTREGA

Archivos sugeridos

OPCIÓN RECOMENDADA	OPCIÓN ACEPTABLE AL INICIO
index.html style.css TEXT	<style> /* CSS interno */ </style> HTML

La recomendada es separar HTML y CSS. Pero si todavía están practicando, se puede aceptar CSS interno.

CRITERIOS

Cómo vamos a revisar el trabajo

HTML

Estructura clara, etiquetas bien usadas y contenido comprensible.

CSS

Reglas correctas, clases claras y propiedades vistas en clase.

Diseño

Buen contraste, jerarquía visual y espaciado suficiente.

Comprensión

Uso correcto de padding, margin, border y box-sizing.

CIERRE

Qué nos tiene que quedar

Hoy trabajamos

- Colores y contraste.
- Tipografías y jerarquía textual.
- Box model: content, padding, border y margin.
- Tarjetas simples con código y resultado visual.

CSS no es decorar: es ordenar visualmente una interfaz.

Próxima clase: Flexbox, Grid y diseño responsive.