

## Profundizando las propiedades CSS

Profundizando las propiedades CSS	1
¿Qué aprenderás?	2
Introducción	2
Modelo de cajas	3
Propiedades del modelo de cajas	3
Modificando elementos desde el inspector	6
Reseteando las propiedades	7
Propiedad Display (inline, block, inline-block)	7
Elementos block	7
Elementos inline	8
Inline-block	8
Width y Height + Unidades de medida	8
Unidades absolutas	9
Unidades relativas	9
Ejercicio guiado: Codo a Codo	10
Resumen	14



**¡Comencemos!**

## ¿Qué aprenderás?

- Implementar estilos CSS, utilizando el modelo de cajas y las propiedades display, unidades de medida y estilos tipográficos avanzados, para la definición de aspectos visuales de una interfaz web.

## Introducción

Hasta ahora hemos visto cómo incluir estilos CSS en nuestros sitios, a través de las propiedades de las fuentes, colores y fondo.

Además, hemos experimentado la importancia del estilo en una página web, pero ¿Qué pasa con su adaptabilidad a distintos dispositivos? Debes conocer al menos una página que no se adecúa correctamente al tamaño de la pantalla de tu celular. Es por esto que es importante que conozcas el concepto del “modelo de cajas” en CSS, que permite anticipar y controlar cómo se comportará el contenido.

**¡Vamos con todo!**



## Modelo de cajas

El modelo de cajas es uno de los conceptos más importantes detrás de CSS, y responde a la pregunta de cuánto mide realmente cada elemento.



Imagen 1. Modelo de cajas.

Fuente: Desafío Latam.

## Propiedades del modelo de cajas

Primero, vamos a analizar el **área de contenido**. Es lo que está comprendido dentro del elemento, como, por ejemplo, textos e imágenes.

Para editarla, podemos alterar el contenido (ya sea agregándole, editando o quitándole contenido en HTML) o asignándole un **ancho (width)** y/o un **alto (height)** como propiedades en CSS.

Además del área de contenido existe el **padding**, que se utiliza para generar espacio alrededor del contenido, dentro del borde del elemento.



El padding es siempre transparente.

El **borde (border)** es lo que contiene y está alrededor del contenido + el padding. Las propiedades de borde de CSS permiten especificar el estilo, el ancho y el color del borde de cada elemento. Aunque, por defecto, en la mayoría de las etiquetas el valor de la propiedad de borde es none, **border: none;**

Por su lado, el **margen (margin)** es espacio alrededor de los elementos, fuera de los bordes definidos. El margen es siempre transparente.

Nosotros podemos cambiar los valores de todas estas propiedades utilizando CSS, pero antes de cambiarlas directamente con código, utilizaremos el inspector de elementos para familiarizarnos con estas propiedades y los valores que vienen por defecto en cada etiqueta.

Si inspeccionamos primero la etiqueta `<body>` de nuestra página, veremos lo computado por el navegador. En la pestaña de **Style**, del inspector de elementos, bajaremos hasta encontrar una caja que dice `margin`, `border` y `padding`.

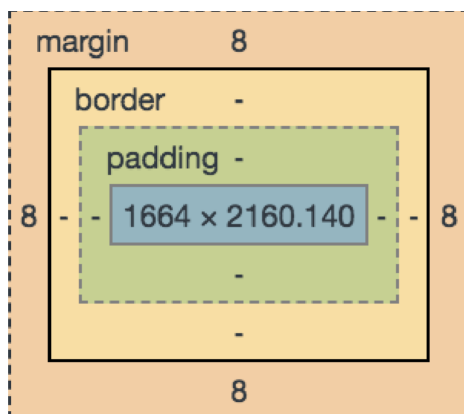


Imagen 2. Propiedades del modelo de cajas.  
Fuente: Desafío Latam.

Ahí podremos ver cuánto mide el área de contenido de nuestro `body` (ancho x alto), el `padding` (que no tiene en este caso), el `border` (que tampoco tiene) y un `margin` que es de 8px en todas las direcciones.

Si inspeccionamos la etiqueta `<nav>` podremos ver que solo tiene un área de contenido delimitada por un ancho x alto. Luego, nos vamos a la etiqueta `<ul>` y vamos encontrando cosas curiosas.

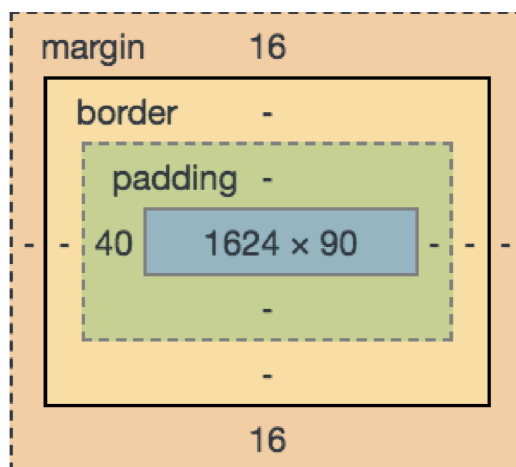


Imagen 3. Propiedades del modelo de cajas.  
Fuente: Desafío Latam.

Podemos observar el tamaño de su área de contenido (ancho x alto), un padding solo a la izquierda de 40px, una carencia de borde y un margen de 16px arriba y abajo.

Luego, vamos a inspeccionar la etiqueta <header>, que al igual que la etiqueta <nav>, solo tiene un área de contenido delimitada por un ancho x alto. Pero, si nos vamos a la etiqueta <h1> y <h2>, podremos ver que tienen propiedades de margen arriba y abajo.

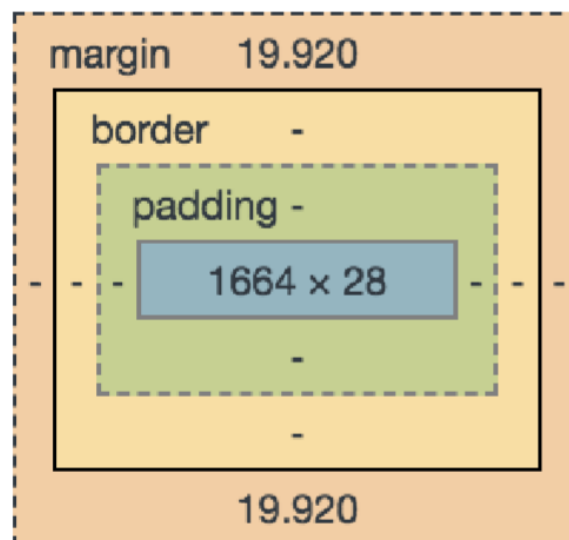
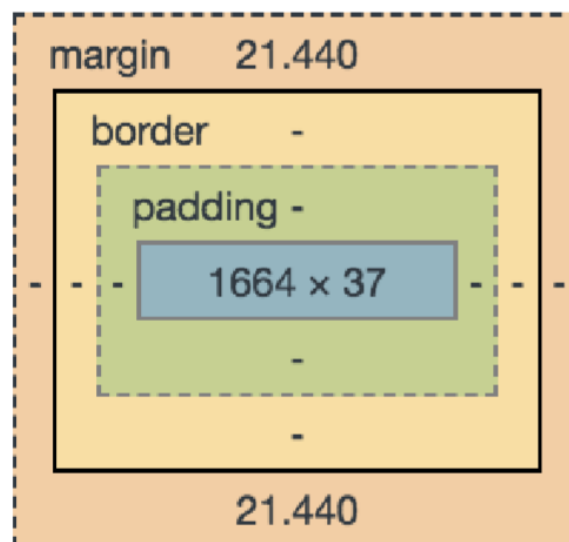


Imagen 4. Propiedades del modelo de cajas - header & nav.  
Fuente: Desafío Latam.

Podemos ir inspeccionando una a una las etiquetas, pero lo importante de este ejercicio es darnos cuenta que cada etiqueta tiene propiedades con valores por defecto.

## Modificando elementos desde el inspector

Desde el inspector de elementos, podremos modificar las características de cada caja. Si vamos a padding o height, podremos ir cambiando valores para verlos reflejados de inmediato en la pantalla. Es importante destacar que podemos alterar cualquier propiedad desde el inspector de elementos, pero estos cambios no quedan guardados en el código. Incluso, puedes intentarlo con cualquier otro sitio que no te pertenezca, de manera online.



Puedes abrir el inspector de elementos con el atajo `ctrl + c` en Windows, o directamente llevando el mouse encima de un elemento y presionando click derecho, seguido de la opción "Inspeccionar"

Hagamos alguna práctica rápida utilizando el Hero Section que creamos en la lectura anterior. Si ocupamos el inspector de elementos para modificar el color de texto del primer encabezado, quedaría de la siguiente manera.



Imagen 5. Modificando elementos con el inspector.

Fuente: Desafío Latam.

Como puedes apreciar, podemos cómodamente hacer modificaciones a nuestro sitio y luego copiar y pegar los estilos declarados en la pestaña Style. Es importante que recuerdes que estos cambios son completamente volátiles, a penas recargues la página se perderán.

## Reseteando las propiedades

Cuando generamos el estilo, debemos tener en consideración que existen propiedades por defecto que tiene cada navegador y que pueden afectar la forma en que vemos el resultado final de los estilos que hemos aplicado.

Es por esto, que se suelen iniciar las propiedades generales por un valor por defecto, antes de comenzar a añadir estilos. Esta técnica se llama “reset CSS”.

Un ejemplo de esto, sería lo siguiente:

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

El \* como selector en CSS significa todo (es el selector universal que aplica a todas las etiquetas), y que esté seguido de body separado por un espacio, significa que a todas las etiquetas que están dentro del body le daremos propiedades en el bloque de declaración.

## Propiedad Display (inline, block, inline-block)

La propiedad display es la propiedad de CSS más importante para controlar la disposición de los elementos.

Cada elemento HTML tiene un valor de visualización predeterminado según el tipo de elemento que sea. El valor de visualización predeterminado para la mayoría de los elementos es en **block** o **inline**.

### Elementos block

Un elemento de bloque siempre comienza en una nueva línea y ocupa todo el ancho disponible (se extiende desde izquierda hacia la derecha todo lo que pueda).

Algunos ejemplos son:

- `<div>`.
- `<h1>` - `<h6>`.
- `<p>`.
- `<form>`.
- `<header>`.
- `<footer>`.
- `<section>`.

## Elementos inline

Por su parte, los elementos inline no comienzan en una nueva línea y su ancho ocupa el mínimo espacio posible (solo lo necesario según su contenido).

Algunos ejemplos son:

- `<span>`.
- `<a>`.
- `<em>`.
- `<strong>`.

## Inline-block

Existe un híbrido de valor display entre el block y el inline. El **inline-block**.

La diferencia principal entre el inline-block y el block es que no agrega un salto de línea después del elemento, por lo que este puede ubicarse junto a otros elementos.

La diferencia principal entre el inline-block y el inline es que permite establecer un ancho (width) y una altura (height) del elemento. Además, respeta los márgenes y paddings. Un ejemplo claro son las imágenes, ya que se despliegan unas al lado de las otras pero podemos asociar un ancho (width) y un alto (height).

## Width y Height + Unidades de medida

**Width** y **height** son las propiedades que se utilizan para establecer el ancho y el alto de un elemento, respectivamente. Las propiedades de anchura y altura no incluye el padding, bordes o márgenes, sino que afectan directamente al área de contenido del elemento.



Los valores que pueden alcanzar estas propiedades son valores de longitud basadas en las unidades de medida estándar para CSS. Podemos clasificar las unidades de medida para CSS en dos, **unidades absolutas** y **unidades relativas**.

## Unidades absolutas

Las unidades de medida absolutas son aquellas que están completamente definidas, es decir, se encuentran definidas en términos concretos y de manera medible. Esto quiere decir que su valor no depende de otro valor de referencia. La que hemos estado utilizando hasta ahora es el pixel, pero también existen los milímetros, centímetros, pulgadas y puntos. Pero, como estamos trabajando con pantallas, lo más lógico sería que trabajemos con los pixeles, que es la unidad mínima de resolución de la pantalla.



Los píxeles (px) son relativos al dispositivo de visualización, por ende cambian si el dispositivo es de una alta resolución o baja.

## Unidades relativas

Las unidades relativas son usadas cuando se quiere conseguir una distribución de elementos en nuestra página que responda activamente a la resolución del dispositivo que se está ocupando para visitar el sitio web, a este concepto en donde los elementos modifican sus medidas en función a la resolución le llamamos "Responsividad".

Estas unidades no son valores exactos, sino que se calculan a partir de otro valor de referencia. A pesar de parecer más difíciles de calcular, son las más utilizadas en el diseño de sitios web responsive por su adaptabilidad a los diferentes dispositivos.

- **Porcentaje (%)**: Su valor está calculado en base a su contenedor. Si, por ejemplo, el porcentaje se utiliza para establecer la anchura de un elemento, su referencia es la anchura de su elemento contenedor. Si el elemento no se encuentra dentro de ningún otro elemento, su referencia es la anchura de la página entera.
- **Viewport width (vw) y viewport height (vh)**: Son medidas relativas de acuerdo al viewport o ventana de visualización del navegador.
  - 1vw = 1% del ancho de la ventana de visualización.
  - 100vw = el ancho total de la ventana de visualización.
  - 1vh = 1% de la altura de la ventana de visualización.
  - 100vh = altura de la ventana de visualización.

- **rem:** Las unidades rem dependen directamente del tamaño de fuente del elemento `<html>`. Si el `font-size` del `<html>` es 16px, 1rem sería igual a 16px en cualquier parte del documento.
- **em:** Las unidades em dependen del font-size definido dentro del elemento en el que se encuentra.

Las últimas dos medidas (rem y em) son más utilizadas para los tamaños de fuente, ahora que vamos a trabajar sobre el ancho (width) y el alto (height) nos fijamos más en los porcentajes (%).



**Antes de continuar:**

- ¿Recuerdas las propiedades del modelo de cajas?
- ¿Puedes definir en tus palabras los elementos block, inline e inline-block?
- ¿Reconoces cuáles son las unidades absolutas y relativas?

Si pudiste responder con facilidad todas las preguntas, puedes seguir adelante, si alguna te presenta dificultad, asegúrate de aclarar los puntos que más te hayan costado.

## Ejercicio guiado: Codo a Codo

A continuación, utilizaremos los conceptos que hemos aprendido en esta lectura para ir maquetar la sección “Próxima Charla” del [Meet & Coffee](#). En donde tenemos una imagen a la izquierda, y un bloque de texto a la derecha centrado verticalmente, además ambos elementos están centrados al medio de la pantalla.

Para la aplicación de CSS utilizaremos el siguiente bloque de elementos HTML.

```
<section id="Proxima-Charla">
  <div class="wrapper">
    
    <div class="texto">
      <h3>Próxima charla</h3>
      <h4>Big Data para Noobs</h4>
      <h6>Big Data para Noobs</h6>
      <span>
        Data Scientist, PhD Theoretical Physics, Carneige Mellon
        University.
      </span>
    </div>
  </div>
```

```
En esta charla revisaremos las técnicas y tecnologías que hacen  
que sea económico hacer frente a los datos a una escala extrema.  
</p>  
</div>  
</div>  
</section>
```

Con solo copiar y pegar este código deberás ver lo que se muestra en la siguiente imagen.



#### Próxima charla

##### Big Data para Noobs

Big Data para Noobs

Data Scientist, PhD Theoretical Physics, Carnegie Mellon University.

En esta charla revisaremos las técnicas y tecnologías que hacen que sea económico hacer frente a los datos a una escala extrema.

Imagen 6. Muestra de elementos HTML sin estilos.  
Fuente: Desafío Latam.

Muy bien, ahora que tenemos los elementos HTML necesarios ¡Comencemos!

- **Paso 1:** Lo primero será definir el color de fondo de la sección, color de texto, un padding de 100px y declarar un ancho con un margen al envoltorio de los elementos para centrarlos. El código CSS quedaría de la siguiente manera.

```
#Proxima-Charla {  
  background: #a4452c;  
  padding: 100px;  
  color: white;  
}  
  
#Proxima-Charla .wrapper {  
  width: 800px;
```

```
margin: auto;  
}
```

Esto hará que nuestra página tome la siguiente forma.

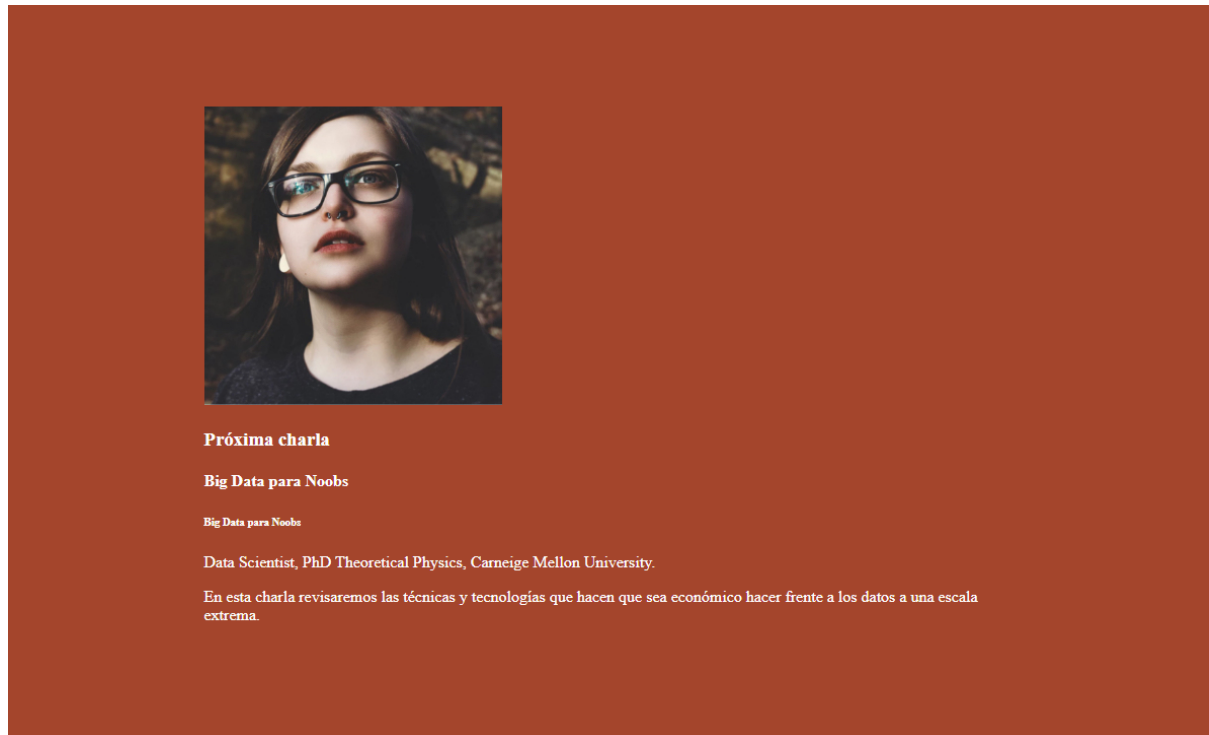


Imagen 7. Muestra de elementos HTML sin estilos.

Fuente: Desafío Latam.

- **Paso 2:** Ahora que tenemos la estética un poco más parecida a la [maqueta](#), pasemos a definir el ancho que tendrá cada parte del envoltorio, es decir la imagen y el **div** con la clase **texto**.

```
.wrapper img {  
  width: 25%;  
}  
  
.wrapper .texto {  
  width: 70%;  
  margin-left: 4%;  
}
```

Seguramente has notado que la suma de porcentajes no dan un 100%, sino que suman un 99%, esto se debe a un pequeño extra que no incluimos para evitar diferencias de algunos navegadores, los cuales agregan algunos píxeles a los elementos de forma interna y descuadran nuestros cálculos.

Al comienzo puede ser un poco frustrante pero no hay de qué preocuparse, este problema hoy día está más que solucionado ocupando las propiedades más actualizadas de CSS como Flexbox o CSS Grid.

Si visualizamos en este momento nuestro documento en el navegador deberemos tener lo siguiente.

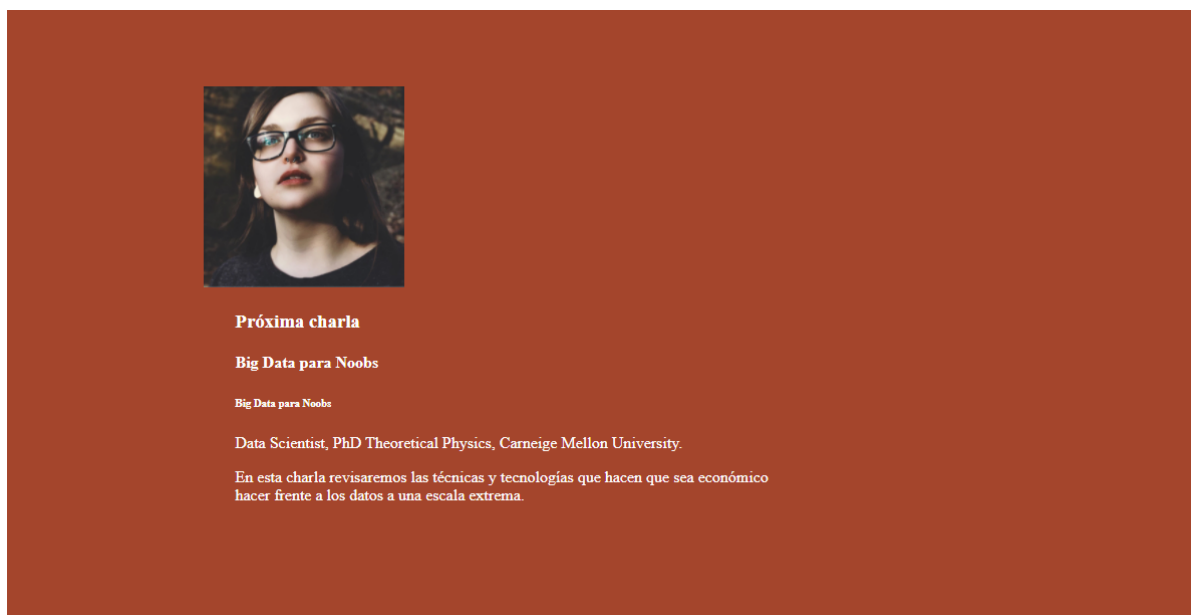



Imagen 8. Elementos con anchos especificados.  
Fuente: Desafío Latam.

- **Paso 3:** Claramente aún no están codo a codo, pero esto lo solucionaremos aplicando un **display inline-block** a ambas partes del envoltorio.

```
.wrapper img, .texto {  
  display: inline-block;  
  vertical-align: middle;  
}
```

Con estos estilos aplicados, deberemos observar como la imagen y el texto se posicionan uno al lado de otro



**Próxima charla**

**Big Data para Noobs**

Big Data para Noobs

Data Scientist, PhD Theoretical Physics, Carnegie Mellon University.

En esta charla revisaremos las técnicas y tecnologías que hacen que sea económico hacer frente a los datos a una escala extrema.

Imagen 9. Elementos ubicados uno al lado del otro.  
Fuente: Desafío Latam.

¡Muy bien! Ahí lo tenemos, ambos elementos ubicados codo a codo en una misma sección.

## Resumen

- El modelo de cajas es uno de los conceptos más importantes detrás de CSS, y responde a la pregunta de cuánto mide realmente cada elemento.
- La técnica “reset CSS” consiste en iniciar las propiedades generales por un valor por defecto, antes de comenzar a añadir estilos.
- La propiedad display es la propiedad de CSS más importante para controlar la disposición de los elementos.
- Un elemento de bloque siempre comienza en una nueva línea y ocupa todo el ancho disponible (se extiende desde izquierda hacia la derecha todo lo que pueda).
- Los elementos inline no comienzan en una nueva línea y su ancho ocupa el mínimo espacio posible (solo lo necesario según su contenido).
- El **inline-block** no agrega un salto de línea después del elemento, por lo que el elemento puede ubicarse junto a otros elementos. Además, permite establecer un ancho (width) y una altura (height) del elemento. Además respeta los márgenes y paddings.
- **Width** y **height** son las propiedades que se utilizan para establecer el ancho y el alto de un elemento, respectivamente.

- Las unidades de medida absolutas son unidades que están completamente definidas, o sea, se encuentran definidas en términos concretos y de manera medible. Esto quiere decir que su valor no depende de otro valor de referencia.
- Las unidades relativas no son valores exactos, sino que se calculan a partir de otro valor de referencia. Son las más utilizadas en el diseño de sitios web responsive por su adaptabilidad a los diferentes dispositivos.