

---

# Producción Digital I

**Lic. Wenceslao Zavala**

11 de Abril 2019

[consultas@producciondigitalup.com](mailto:consultas@producciondigitalup.com)

[www.producciondigitalup.com](http://www.producciondigitalup.com)

---

# Trabajo Práctico N°2

## Maqueta HTML

A partir del boceto pensado para la home, realizar el maquetado en HTML de la versión de escritorio del Wireframe entregado en el trabajo anterior.

Realizar la maqueta en HTML de la versión de escritorio del Wireframe entregado en el trabajo anterior. El tamaño de pantalla de base para computadoras es de 1300 pixeles como ancho máximo.

Y realizar también los wireframes/bocetos de la home adaptada a tabletas (768px) y celulares (480px).

La entrega consta de 2 archivos para la maqueta HTML, 1 archivo de extensión HTML, y otro de CSS, no deben haber reglas de CSS dentro del archivo HTML; y de otro archivo en PDF con los wireframes.

---

**Fecha de entrega:**

Clase N° 3 (04/04/19)

# HTML + CSS

**<contenidos>**

**apariciencia }**

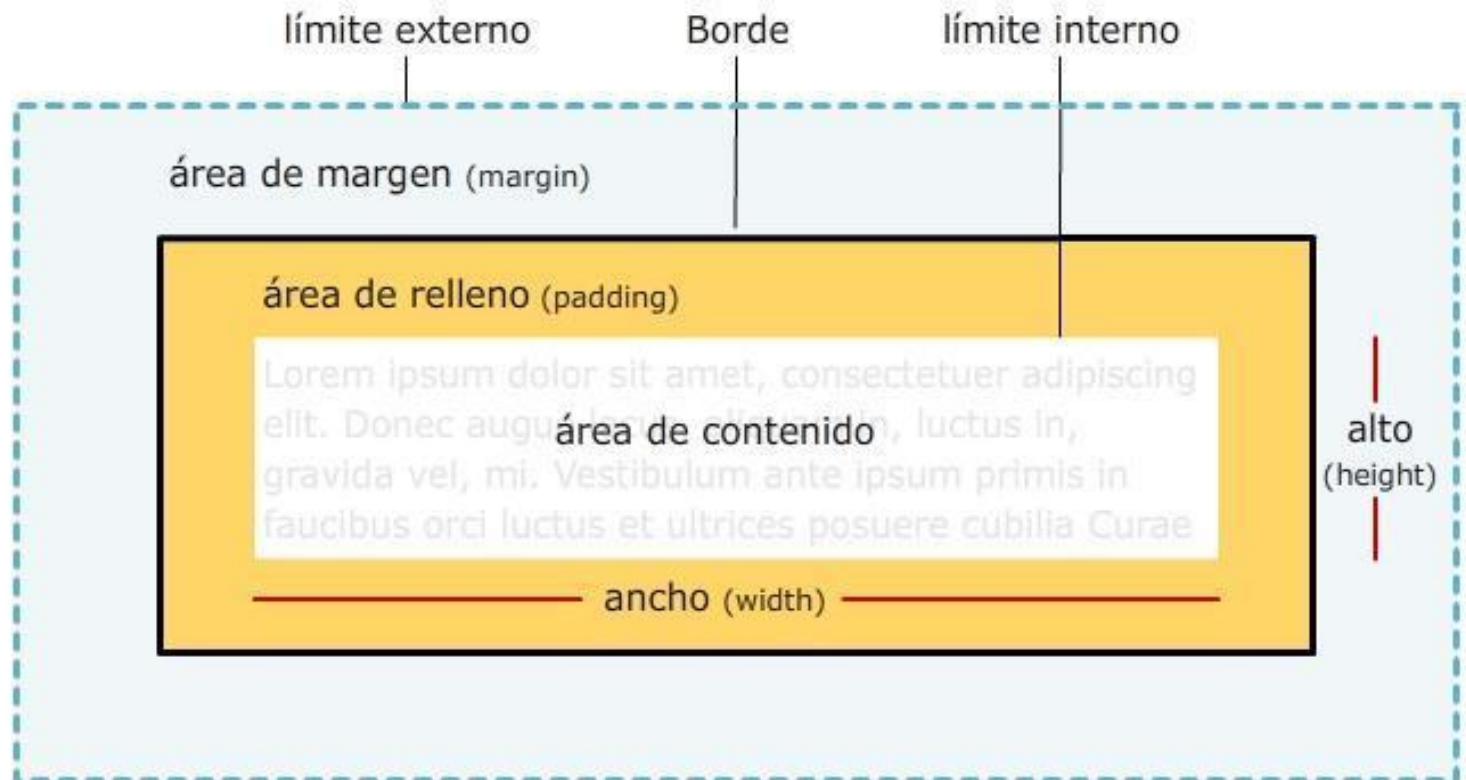
—  
**HTML**

**Hyper Text Mark-up  
Language**

**Lenguaje de Etiquetas  
de Hipertexto**

---

# Caja de elemento HTML



—

# CSS

## Cascading Style Sheet

## Hojas de Estilo en Cascada

---

# Tipos de Reglas (selectores) CSS

**Etiquetas HTML [TAG]** modifica todas las instancias de la etiqueta, como p, h1 o DIV.

Es el nombre es la etiqueta HTML, y no se necesita atributo para aplicarlo

**Clase [CLASS]** puede ser aplicado en múltiples elementos dentro de un mismo HTML.

todos los nombres de las reglas de clase empiezan por un punto (.)

usa el atributo **class** para aplicarlo

**ID [ID o DIV]** aplicables a un sólo elemento dentro del HTML.

el nombre de la regla comienza con #

usa el atributo **id** para aplicarlo

---

---

# CSS Reset

Se deben normalizar los estilos CSS antes de empezar.

Los navegadores tienen diferentes propiedades predeterminadas para los elementos, por eso se debe hacer que todos los elementos tengan el mismo aspecto.

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

---



---

# Wireframe HTML

## Encabezado

(lo primero que quiero que vea el usuario)

## Contenido

(seguido del contenido)

## Pie de Página

(en general la información menos importante)

---



CAREERS AT PIXAR



TRAILER



BAO



#header

#nav

#contenido

.noticia

.noticia

.noticia

.noticia

.noticia

.noticia

#footer

#header

#nav

#contenido

.noticia

.noticia

.noticia

.noticia

.noticia

.noticia

#footer

---

# Elemento de Línea o Bloque

El HTML clasifica a todos sus elementos en dos grupos:

- **Elementos de bloque** ("**block** elements" en inglés) siempre empiezan en una nueva línea y ocupan todo el ancho disponible hasta el final de la línea
  - **Elementos en línea** ("**inline** elements" en inglés) no empiezan necesariamente en nueva línea y sólo ocupan el espacio necesario para mostrar sus contenidos
-

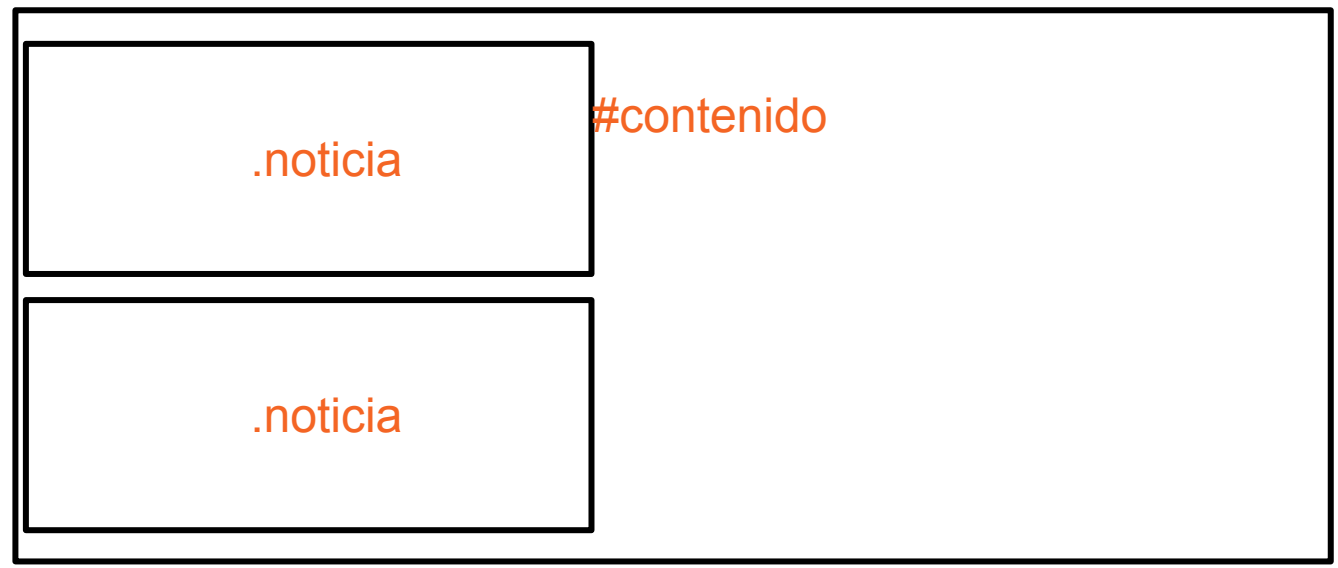
---

# Elemento Bloque

Casi todas las etiquetas de HTML son elementos de Bloque.

No importa el valor del ancho que tengan, ocupan todo el ancho de la caja que la contiene.

```
<body>  
<div>  
<h1>  
<h2>  
<h3>  
<p>
```



---

# Elemento en Línea

Los elementos en línea normalmente sólo pueden contener texto y otros elementos en línea. Se visualizan sin comenzar en una nueva línea.

<img>

<a>

<strong>

<em>



#contenido

Lorem ipsum dolor sit amet,  
[consectetur](#) adipiscing elit.

---

# Propiedad Display

Es la propiedad más importante para controlar las estructuras.

Cada elemento tiene un valor de **display** por predeterminado dependiendo de qué tipo de elemento sea.

El valor por defecto para la mayoría de los elementos es usualmente **block** (de **bloque**) o **inline** (en **línea**).

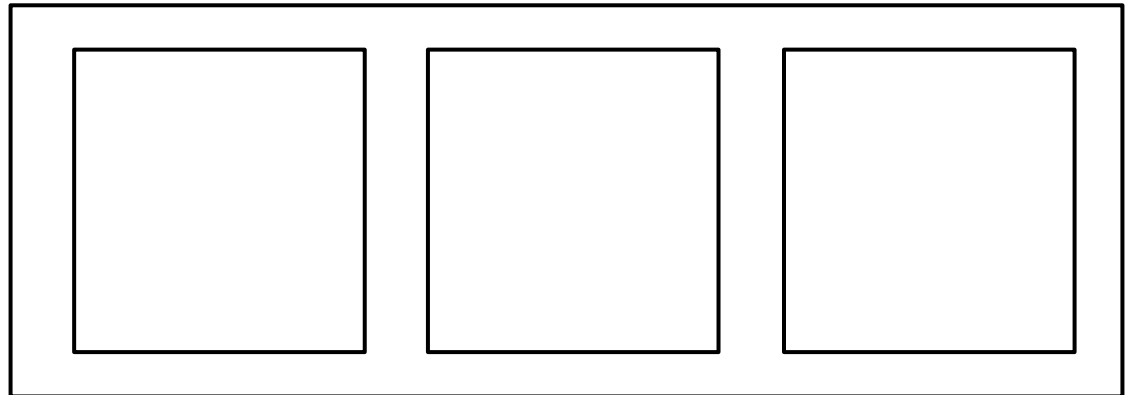
---



---

# Layout de 3 Columnas

```
<div class="section">  
  <div class="columna"></div>  
  <div class="columna"></div>  
  <div class="columna"></div>  
</div>
```



---

# Método Float

```
.columna {  
    float: left;  
    width: 33.3%;  
}
```

```
/* Limpia la propiedad floats después de las  
columnas */
```

```
.section:after {  
    content: "";  
    display: block;  
    clear: both;  
}
```

---

#header

#nav

#contenido

.noticia

.noticia

.noticia

.noticia

.noticia

.noticia

#footer

---

# Float & Clear

La propiedad **Float** hace que cualquier elemento de bloque, se comporte como elemento de línea.

<h1>

<p>

<p>

Para romper con esa herencia se debe limpiar ese comportamiento con la propiedad **Clear**.

<h1>

<p>

<p>

---

# Limpiar Herencia Float desde CSS

La **propiedad Float** lo convierte a elemento de línea haciendo que su altura sea **igual a 0**.

Para romper con esa herencia se debe limpiar con la **propiedad Clear**. Esto sólo soluciona a los elementos a continuación.

La mejor opción, es crear una regla de tipo **pseudo-selector** para la caja que contiene al elemento:

```
#cajacontenedora:after {  
    clear: both;  
    display: block;  
    content: "";  
}
```

---

# Identificar la altura de elementos

Si un elemento es más alto que el elemento que lo contiene, y se convierte a flotante (float) se desborda fuera de su contenedor.

```
#cajacontenedora:after {  
    clear: both;  
    display: block;  
    content: ""  
}
```

# Barra de Navegación

Barra de navegación  
=  
Lista de vínculos

Una barra de navegación es básicamente una lista de vínculos, por eso es el uso de los elementos `<ul>` y `<li>`

```
<ul>  
  
  <li>  
    <a href="index.html">Home</a>  
  </li>  
  
  <li>  
    <a href="noticias.html">Noticias</a>  
  </li>  
  
  <li>  
    <a href="contacto.html">Contacto</a>  
  </li>  
  
</ul>
```

`<ul>` Unorder List | `<li>` List

---

# Reglas Específicas

Son reglas que le dan una ruta de pertenencia al elemento de HTML para definir propiedades específicas.

```
#nav ul li a {color:#FF00FF;}
```

Sólo para los links que se encuentren en la barra de navegación que estén dentro de una lista. Esto no aplica a los demás vínculos del HTML



---

# Remove las viñetas (bullets)

Son reglas que le dan una ruta de pertenencia al elemento de HTML para definir propiedades específicas.

```
#nav ul {  
    list-style-type: none;  
}
```

**list-style-type: none;** Elimina las viñetas (bullets).

La barra de navegación no necesita marcadores de lista

---

# Barra de Navegación Horizontal

Existen 2 formas para crear una barra horizontal:

Usando el **método inline**

```
li { display:inline;} li a { display:inline-block;}
```

Usando el **método floating**

```
li { float:left ;} li a { display: block;}
```

Ambos métodos sirven, pero:

Para links de un **mismo tamaño** se debe usar el **método floating**. Este elemento no reconoce altura de la caja.

Para un **menú centrado** se debe usar el **método Inline**. Pero no se le puede determinar un ancho a la caja.

---

---

# Remover las viñetas (bullets)

```
li { float: left; }
```

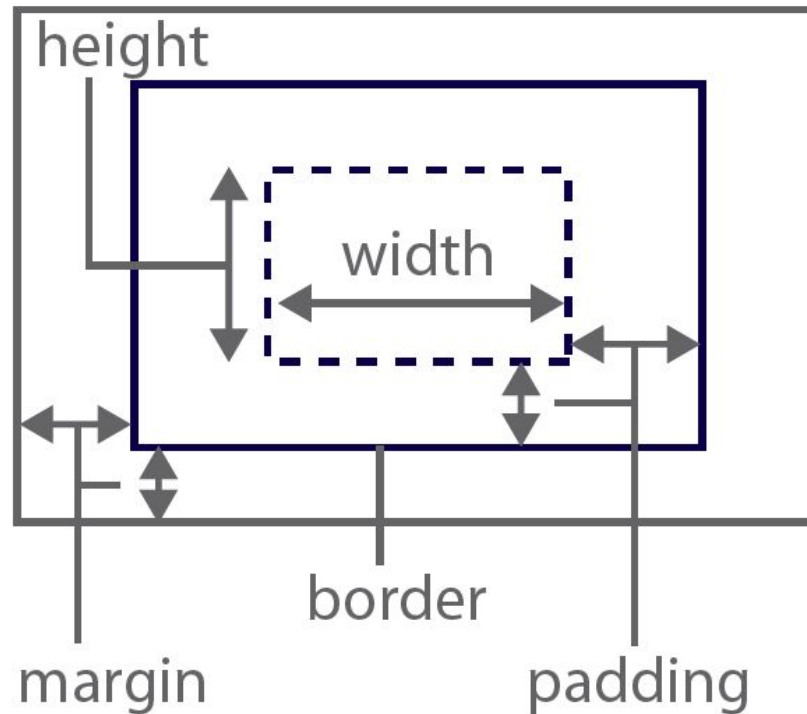
```
a { display: block; }
```

**float: left;** **float** se usa para hacer que los elementos de bloque se ubiquen uno al lado del otro.

**display: block;** Muestra los vínculos como elementos de bloque haciendo que toda el área del enlace sea sensible al clic (no solo el texto) y nos permite especificar el relleno (y la altura, el ancho, los márgenes, etc., si lo desea)

---

# Propiedades CSS de Caja



height; width;  
margin-top;  
margin-right;  
margin-bottom;  
margin-left;  
padding-top;  
padding-right;  
padding-bottom;  
padding-left;

---

---

# Identificar la altura de elementos

Si un elemento es más alto que el elemento que lo contiene, y se convierte a flotante (float) se desborda fuera de su contenedor.

```
#cajacontenedora:after {  
    clear: both;  
    display: block;  
    content: ""  
}
```

---

# Seudo-clases y seudo-elementos

Son reglas de CSS que se utiliza para añadir propiedades a un estado de un elemento

Sintaxis de las pseudo-classes:

```
selector:pseudo-clase { propiedad:valor; }
```

El más utilizado es para el **Roll-Over**

```
a:hover {color:#FF00FF;} /* link mouse over */
```

#header

#nav

#contenido

.noticia

.noticia

.noticia

.noticia

.noticia

.noticia

#footer

---

# max-width - Propiedad CSS

## Definición y uso

La propiedad **max-width** se utiliza para establecer el ancho máximo de un elemento.

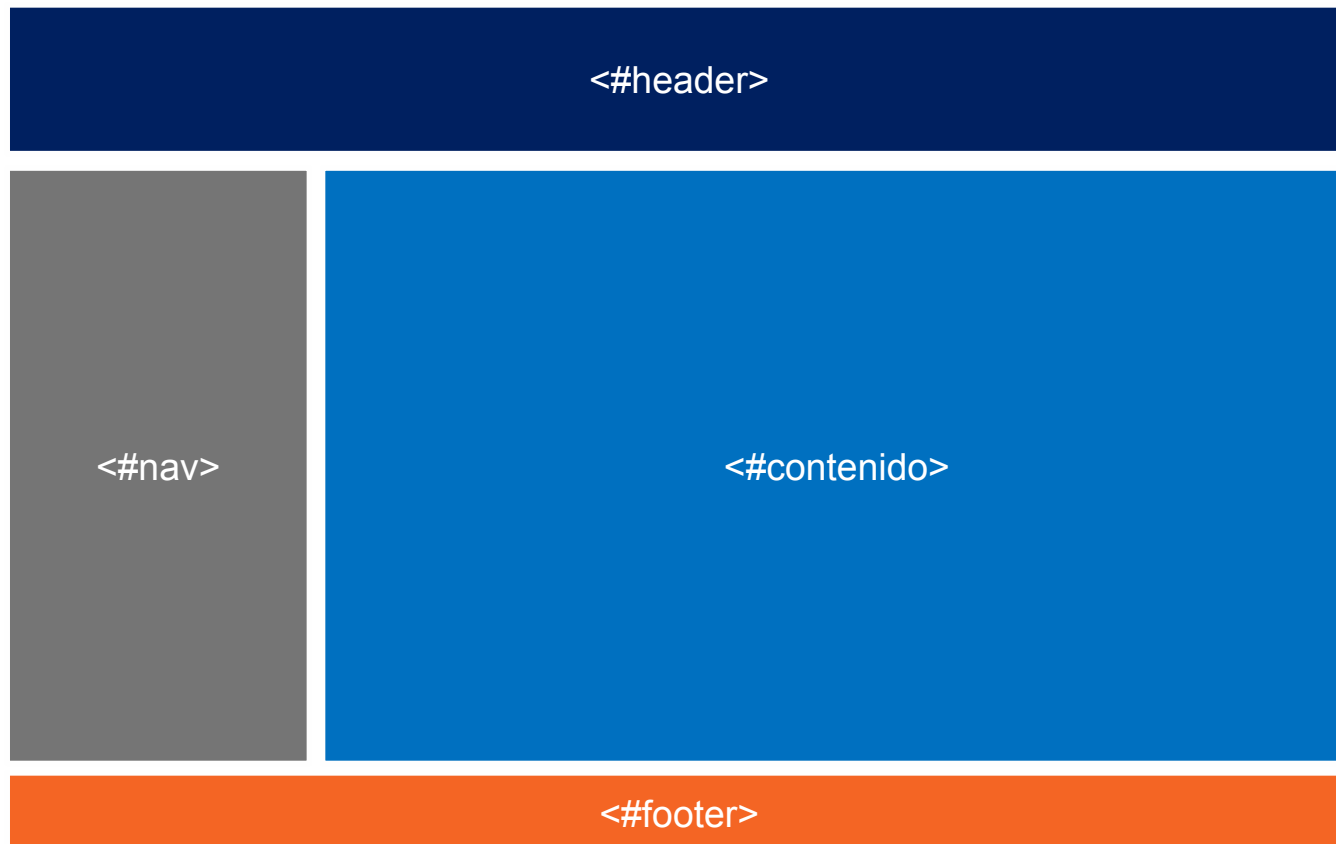
Esto evita que el elemento no supere un tamaño de ancho específico:

```
{ max-width: 1100px;}
```

---



# Cómo Maquetar con DIVs



# Contenedor/envoltura



---

# Adaptable a Diferentes Dispositivos



---

# CSS MediaQueries

**Desktop 1300px**

**Tablets 768px**

```
@media (max-width: 768px) { }
```

**Smartphone 480px**

```
@media (max-width: 480px) { }
```

---

# Viewport

Es el área visible del usuario de una página web

La misma varía con el dispositivo

Para solucionar esta variedad de tamaños, los navegadores de esos dispositivos redujeron la escala de toda la página web para adaptarse a la pantalla.

Con HTML5 se permite controlar el tamaño de la ventana mediante la etiqueta `<meta>`

Debe incluir el siguiente elemento de ventana `<meta>` en todas sus páginas web:

```
<meta name = "viewport" content =  
"width = device-width, initial-scale =  
1.0">
```

---

---

# Modelado de Caja

Cuando se ajusta la propiedad **width** de un elemento, puede parecer más grande de lo que es.

Las propiedades de **width** (ancho) **border** (borde) y **padding** (relleno) estiran el elemento más allá del ancho especificado.

---

# Propiedades CSS3

## Modelado de Caja box-sizing

Sirve para ajustar la caja así el padding y border no incrementan su ancho.

**box-sizing: border-box;**

## Ancho máximo de caja:

`max-width: 1100px;`

## Bordes redondeados:

`border-radius: 6px;`

## Sombra de Cajas:

`box-shadow: 2px 2px 2px #333;`

---

---

# Posicionamiento

Las propiedades de posicionamiento permiten colocar un elemento sin importar el flujo del HTML.

**Static.** Predeterminado.

Está siempre posicionado de acuerdo con el flujo normal de la página.

**Fixed** - Se coloca en relación con la ventana del navegador.

**Relative** - Situado con relación a su posición normal

**Absolute** - Posicionado en relación con el primer elemento padre que tiene una posición que no sea estática. Si no se encuentra dicho elemento, el bloque de contención es <html>



# Trabajo Práctico

## N°3

### Sitio Navegable entre secciones

A partir de la maqueta de HTML realizada en el TP N° 2, se deberá tener el sitio con la navegabilidad a todas las secciones principales, pudiendo identificar con las etiquetas que correspondan a los contenidos de cada pantalla.

Es decir, se deben realizar los archivos de la página de inicio (index.html) y de las secciones principales del sitio. El mismo debe navegar desde la página de inicio a las secciones presentadas y viceversa. El sitio tendrá sólo una hoja de estilo (ej. estilos.css). Todo elemento que tenga un vínculo debe tener el comportamiento de Roll-Over.

**A ningún archivo le debe faltar la etiqueta TITLE, acorde a donde se encuentre el usuario.**

El ancho máximo del sitio no debe superar los 1100px de ancho, y no debe aparecer la barra de desplazamiento (scroll) horizontal.

---

**Fecha de entrega:**

Clase N° 5 (25/04/19)

---

# Producción Digital I

**Lic. Wenceslao Zavala**

11 de Abril 2019

[consultas@producciondigitalup.com](mailto:consultas@producciondigitalup.com)

[www.producciondigitalup.com](http://www.producciondigitalup.com)

---