


| | |
|---|---|
|  | <p style="text-align: center;">UNIVERSIDAD DON BOSCO FACULTAD DE INGENIERÍA ESCUELA DE COMPUTACION</p> |
| <p>CICLO 02-2023</p> | <p style="text-align: center;">GUIA DE LABORATORIO Nº 3</p> |
| | <p>Nombre de la practica: Modelo de caja, posicionamiento, visualización y maquetación de sitios web fijos y fluidos Lugar de ejecución: Laboratorio de Informática Materia: Lenguajes Interpretados en el Cliente</p> |

I. Objetivos

Que el estudiante sea capaz de:

1. Crear páginas web utilizando el Modelo de Cajas, identificando las respectivas partes (margin, border, padding y content).
2. Utilizar los diferentes tipos de posicionamiento CSS y diferenciarlos apropiadamente.
3. Desarrollar páginas web con diseños fijos y fluidos.

II. Introducción Teórica

El modelo de formato visual de CSS (es decir, el modelo que siguen los navegadores para procesar el contenido de un documento para los medios visuales) permite que esas herramientas estén disponibles también para el diseño de las páginas Web. Este modelo de formato visual rige el comportamiento de las cajas generadas por los elementos de la página.

Las hojas de estilo son procesadas por el navegador del usuario siguiendo un determinado modelo de formato visual. Este modelo es conocido como modelo de cuadro o caja, en el que cada elemento de la página genera una o más cajas rectangulares invisibles.

Luego es necesario saber que existen tres tipos de elementos en HTML:

Elementos de bloque

Son aquellos tratados visualmente como bloques separados de los elementos que lo rodean (por ejemplo: <P> o <DIV>). Para hacerlo más sencillo, podemos decir que son aquellos que comienzan una nueva línea dentro del documento. Los elementos a nivel de bloque generan una caja de bloque principal que sólo contiene otras cajas de bloque.

Elementos de línea

Son aquellos que no forman nuevos bloques de contenido; el contenido es distribuido a nivel de las líneas (por ejemplo: o).

Elementos de lista

Son elementos de bloque que generan una caja principal y otras cajas adicionales (generalmente contienen una viñeta o caracteres alfanuméricos) que se agregan al costado del elemento.

En el siguiente ejemplo se muestran los distintos tipos de elementos:

Los Párrafos son elementos de Bloque

Los Enlaces son elementos de Línea

Dentro de un Párrafo, los enlaces siguen siendo elementos de línea

Lista:

1. Elemento1
2. Elemento2
3. ElementoN

En la mayoría de los casos, CSS funciona igual para las cajas inline y para las cajas block-level. Se puede aplicar estilo a la fuente, color o fondo y añadir bordes a ambos tipos de cajas. Sin embargo cuando se trata de márgenes y relleno, los navegadores tratan a las cajas inline de manera diferente.

Mientras que se puede añadir espacio a la izquierda o a la derecha de un elemento inline utilizando relleno a la izquierda o a la derecha o márgenes a la izquierda o la derecha, no se puede aumentar el alto del elemento inline con relleno o márgenes superior o inferior.

El modelo de cajas o Box Model de las hojas de estilo (CSS) describe las cajas rectangulares que son generadas por los elementos HTML. El elemento raíz del documento web, HTML o, mejor aún, BODY dentro de HTML, genera una caja que actúa como contención de las cajas generadas subsecuentemente. Del mismo modo, cada caja generada, puede actuar como bloque de contención de otras cajas generadas por sus elementos descendientes. La utilidad del modelo es que sirve para tener control de las propiedades del elemento y de la caja generada por el mismo. Lo cual permitirá componer visualmente la página web, de un modo más rico y flexible que con HTML puro. Las cajas generadas están compuestas de un área de contenido, un espacio de relleno alrededor de ésta (opcional) denominado padding, un borde exterior al relleno, denominado border, que puede ser visible o no; y un espacio invisible alrededor del borde, denominado margin. Cada una de estas está subdividida en segmentos Top, Bottom, Right y Left.



Propiedades del margen (Margin).

Las propiedades del margen se pueden dividir en cuatro partes: margin-left, margin-right, margin-top y margin-bottom. Estas propiedades determinan el ancho del margen izquierdo, derecho, superior e inferior respectivamente. Los valores que pueden asignarse a estas propiedades pueden especificarse en una unidad de medida, en porcentaje o dejar que el navegador lo determine de forma automática.

Propiedad Extendida

| 'margin-top', 'margin-right', 'margin-bottom', 'margin-left' | |
|--|---|
| Valores: | <ancho-del-margen> inherit |
| Valor inicial: | 0 |
| Se aplica a: | todos los elementos |
| Se hereda?: | no |
| Porcentajes: | se refieren al ancho del bloque de contención |

Propiedad Resumen

| 'margin' | |
|----------------|---|
| Valores: | <ancho-del-margen> {1,4} inherit |
| Valor inicial: | no definido para las propiedades resumidas |
| Se aplica a: | todos los elementos |
| Se hereda?: | no |
| Porcentajes: | se refieren al ancho del bloque de contención |

Margin acepta valores negativos.

El uso de {1,4} indica que pueden especificarse de uno a cuatro valores para esta propiedad, interpretándose de la siguiente forma:

- Un solo valor: Se aplica hacia todos los lados, margin: 10px;
- Dos valores: Los márgenes superior e inferior son determinados por el primer valor y los márgenes derecho e izquierdo son determinados por el segundo valor. margin: 10px 10px;
- Tres valores: El margen superior es definido por el primer valor, el izquierdo y el derecho son definidos por el segundo valor y el margen inferior es definido por el tercer valor. margin: 10px 10px 10px;
- Cuatro valores: Se aplica de forma individual cada valor a los márgenes en el siguiente orden. El primer valor al margen superior, el segundo al margen izquierdo, el tercero al margen inferior y el cuarto valor al margen derecho. margin: 10px 10px 10px 10px;

Propiedades del relleno (Padding).

Las propiedades del relleno especifican el ancho del área de relleno de una caja y pueden dividirse en cuatro partes, al igual que el margen, estas son: padding-left, padding-right, padding-top y padding-bottom. Dichas propiedades determinan el ancho del área de relleno a la izquierda, a la derecha, arriba y abajo del área de contenido, respectivamente. Sus valores pueden ser establecidos en una unidad de medida, en porcentaje. Los valores para estas propiedades no pueden ser negativos. La siguiente tabla muestra el resumen de las propiedades de relleno:

Propiedad Extendida

| 'padding-top', 'padding-right', 'padding-bottom', 'padding-left' | |
|--|---|
| Valores: | <ancho-del-relleno> inherit |
| Valor inicial: | 0 |
| Se aplica a: | todos los elementos |
| Se hereda?: | no |
| Porcentajes: | se refieren al ancho del bloque de contención |

Propiedad Resumen

| 'padding' | |
|----------------|---|
| Valores: | <ancho-del-relleno> {1,4} inherit |
| Valor inicial: | no definido para las propiedades resumidas |
| Se aplica a: | todos los elementos |
| Se hereda?: | no |
| Porcentajes: | se refieren al ancho del bloque de contención |

La interpretación de los valores proporcionados son exactamente igual que para el atributo margin.

Propiedades del borde (Border).

Las propiedades del borde especifican el ancho, color y estilo de una caja. Estas propiedades se aplican a todos los elementos HTML.

Ancho del borde.

Estas propiedades determinan el ancho de cada uno de los bordes que rodea una caja. Es decir, el borde izquierdo, derecho, superior e inferior de la caja. Los valores que pueden asignarse a estas propiedades son:

- thin: para un borde fino.
- medium: para un borde medio.
- thick: para un borde grueso.

También puede establecerse una medida específica utilizando valores en alguna unidad de medida válida. Por ejemplo, pixeles.

Propiedad Extendida

| 'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width' | |
|--|-----------------------------|
| Valores: | <ancho-del-borde> inherit |
| Valor inicial: | medium |
| Se aplica a: | todos los elementos |
| Se hereda?: | no |
| Porcentajes: | no admitidos |

Propiedad Resumen

| 'border-width' | |
|----------------|-----------------------------------|
| Valores: | <ancho-del-borde> {1,4} inherit |
| Valor inicial: | ver las propiedades individuales |
| Se aplica a: | todos los elementos |
| Se hereda?: | no |
| Porcentajes: | no admitidos |

Color del borde.

Con estas propiedades se pueden establecer de forma individual el color del borde izquierdo, derecho, superior e inferior. Las propiedades son: border-left-color, border-right-color, border-top-color y border-bottom-color.

Propiedad Extendida

| 'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color' | |
|--|----------------------------------|
| Valores: | <color> inherit |
| Valor inicial: | el valor de la propiedad 'color' |
| Se aplica a: | todos los elementos |
| Se hereda?: | no |
| Porcentajes: | no admitidos |

Propiedad Resumen

| 'border-color' | |
|----------------|---------------------------------------|
| Valores: | <color> {1,4} transparent inherit |
| Valor inicial: | ver las propiedades individuales |
| Se aplica a: | todos los elementos |
| Se hereda?: | no |
| Porcentajes: | no admitidos |

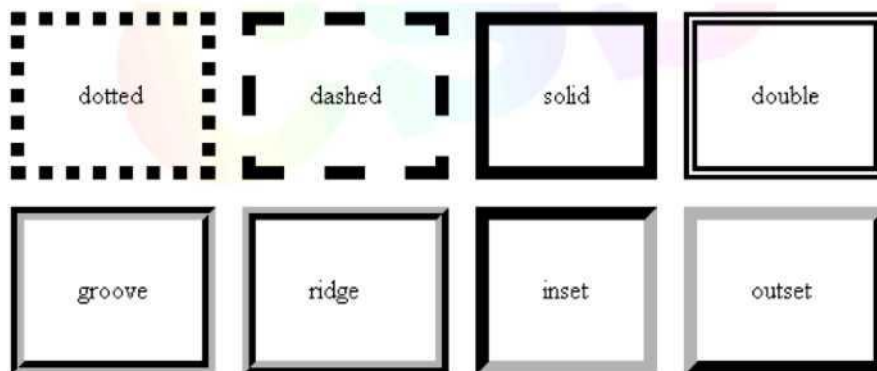
Estilo del borde.

Estas propiedades especifican el estilo de la línea del borde de una caja. Esta puede ser sólida, doble, punteada, etc. Estas propiedades son: `border-left-style`, `border-right-style`, `border-top-style` y `border-bottom-style`.

Los valores que pueden ser asignados a las propiedades de borde son los siguientes:

- `none`: que indica que la caja aparecerá sin borde. Esto obliga a que el valor para la propiedad `border-width` sea cero.
- `hidden`: produce el mismo efecto que `none`. La caja aparece sin borde.
- `dotted`: el borde es una serie de puntos.
- `dashed`: el borde es una serie de pequeños segmentos de línea.
- `solid`: el borde es un único segmento de línea sólido.
- `double`: el borde son dos líneas sólidas. La suma de las dos líneas y el espacio entre ellas es igual al valor de `border-width`.
- `groove`: el borde luce como si estuviera tallado en la página.
- `ridge`: el borde aparece como si estuviera sobresaliendo de la página.
- `inset`: el borde hace que la caja luzca como si estuviera empotrada en la página.
- `outset`: el borde hace que la caja luzca como si sobresaliera de la página.

La forma que cada uno de estos valores le da al borde de las cajas, marcos o cuadros se muestra en la siguiente figura:



| Propiedad Extendida | | Propiedad Resumen | |
|--|------------------------------|-------------------|--------------------------------------|
| 'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-style' | | 'border-style' | |
| Valores: | <estilo-del-borde> inherit | Valores: | < estilo-del-borde > {1,4} inherit |
| Valor inicial: | none | Valor inicial: | ver las propiedades individuales |
| Se aplica a: | todos los elementos | Se aplica a: | todos los elementos |
| Se hereda?: | no | Se hereda?: | no |
| Porcentajes: | no admitidos | Porcentajes: | no admitidos |

Existen dos grandes clasificaciones de los elementos HTML que distingue un navegador:

- **Elementos de bloque**

Los elementos de bloque ("block elements" en inglés) siempre empiezan en una nueva línea, ocupan todo el espacio disponible hasta el final de la línea y pueden contener otros elementos de bloque o de línea.

- **Elementos de línea**

Los elementos de línea ("inline elements" en inglés) no empiezan necesariamente en nueva línea, sólo ocupan el espacio necesario para mostrar sus contenidos, y solo pueden contener otros elementos de línea o contenido.

Algunos ejemplos de elementos de bloque son: *header, nav, aside, section, article, div, p, footer, h1, h2, h3, h4, h5, h6*, etc.

Por su parte, algunos ejemplos de elementos de línea son: *a, b, i, br, cite, img, span, sup, sub*, etc.

Posicionamiento

Los navegadores crean y posicionan de forma automática todas las cajas que forman cada página HTML. No obstante, CSS permite al diseñador modificar la posición en la que se muestra cada caja.

Utilizando las propiedades que proporciona CSS para alterar la posición de las cajas es posible realizar efectos muy avanzados y diseñar estructuras de páginas que de otra forma no serían posibles.

El estándar de CSS define cinco modelos diferentes para posicionar una caja:

- **Posicionamiento normal o estático.** Se trata del posicionamiento que utilizan los navegadores si no se indica lo contrario.
- **Posicionamiento relativo:** variante del posicionamiento normal que consiste en posicionar una caja según el posicionamiento normal y después desplazarla respecto de su posición original.
- **Posicionamiento absoluto:** la posición de una caja se establece de forma absoluta respecto de su elemento contenedor y el resto de elementos de la página ignoran la nueva posición del elemento.

- **Posicionamiento fijo:** variante del posicionamiento absoluto que convierte una caja en un elemento inamovible, de forma que su posición en la pantalla siempre es la misma independientemente del resto de elementos e independientemente de si el usuario sube o baja la página en la ventana del navegador.
- **Posicionamiento flotante:** se trata del modelo más especial de posicionamiento, ya que desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.

Propiedad position

El posicionamiento de una caja se establece mediante la propiedad *position*:

| position | Posicionamiento |
|----------------------|--|
| Valores | <code>static relative absolute fixed inherit</code> |
| Se aplica a | Todos los elementos |
| Valor inicial | <code>static</code> |
| Descripción | Selecciona el posicionamiento con el que se mostrará el elemento |

El significado de cada uno de los posibles valores de la propiedad *position* es el siguiente:

- **static:** corresponde al posicionamiento normal o estático. Si se utiliza este valor, se ignoran los valores de las propiedades *top*, *right*, *bottom* y *left* que se verán a continuación.
- **relative:** corresponde al posicionamiento relativo. El desplazamiento de la caja se controla con las propiedades *top*, *right*, *bottom* y *left*.
- **absolute:** corresponde al posicionamiento absoluto. El desplazamiento de la caja también se controla con las propiedades *top*, *right*, *bottom* y *left*, pero su interpretación es mucho más compleja, ya que el origen de coordenadas del desplazamiento depende del posicionamiento de su elemento contenedor.
- **fixed:** corresponde al posicionamiento fijo. El desplazamiento se establece de la misma forma que en el posicionamiento absoluto, pero en este caso el elemento permanece inamovible en la pantalla.

La propiedad *position* no permite controlar el posicionamiento flotante, que se establece con otra propiedad llamada *float* y que se explica más adelante. Además, la propiedad *position* sólo indica cómo se posiciona una caja, pero no la desplaza.

Propiedades de desplazamiento

Normalmente, cuando se posiciona una caja también es necesario desplazarla respecto de su posición original o respecto de otro origen de coordenadas. CSS define cuatro propiedades llamadas *top*, *right*, *bottom* y *left* para controlar el desplazamiento de las cajas posicionadas:

| | |
|--|--|
| top right bottom left | Desplazamiento superior Desplazamiento lateral derecho Desplazamiento inferior Desplazamiento lateral izquierdo |
| Valores | <medida> <porcentaje> auto inherit |
| Se aplica a | Todos los elementos posicionados |
| Valor inicial | auto |
| Descripción | Indican el desplazamiento horizontal y vertical del elemento respecto de su posición original |

En el caso del posicionamiento relativo, cada una de estas propiedades indica el desplazamiento del elemento desde la posición original de su borde superior/derecho/inferior/izquierdo. Si el posicionamiento es absoluto, las propiedades indican el desplazamiento del elemento respecto del borde superior/derecho/inferior/izquierdo de su primer elemento padre posicionado.

En cualquiera de los dos casos, si el desplazamiento se indica en forma de porcentaje, se refiere al porcentaje sobre la anchura (propiedades right y left) o altura (propiedades top y bottom) del elemento.

Visualización

Además de las propiedades que controlan el posicionamiento de los elementos, CSS define otras cuatro propiedades para controlar su visualización: **display**, **visibility**, **overflow** y **z-index**.

Utilizando algunas de estas propiedades es posible ocultar y/o hacer invisibles las cajas de los elementos, por lo que son imprescindibles para realizar efectos avanzados.

- **Propiedad display**

Es la primera de las propiedades de visualización en CSS y es una de las más útiles para diseñadores.

| | |
|----------------|---|
| display | Visualización de un elemento |
| Valores | inline block none list-item run-in inline-block table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption inherit |
| Se aplica a | Todos los elementos |
| Valor inicial | inline |
| Descripción | Permite controlar la forma de visualizar un elemento e incluso ocultarlo |

Los valores más utilizados son inline, block , none e inline-block.

- ✓ El valor **block** permite mostrar un elemento como si fuera un elemento de bloque, independientemente del tipo de elemento que se trate.
- ✓ El valor **inline** permite visualizar un elemento en forma de elemento en línea, independientemente del tipo de elemento que se trate.

- ✓ El valor `none` permite ocultar un elemento y hacer que desaparezca de la página. El resto de elementos se visualizan como si no existiera el elemento oculto, es decir, pueden ocupar el espacio en el que se debería visualizar el elemento.
- ✓ El valor `inline-block` es muy interesante ya que permite obtener un comportamiento intermedio entre los elementos de línea y los de bloque. **Los elementos inline-block fluyen con el texto y demás elementos** como si fueran elementos en-línea **y además respetan el ancho, el alto y los márgenes verticales**. Son lo mejor de los dos mundos. Con este valor se pueden conseguir comportamientos similares que con el uso de la propiedad `float` pero sin el dolor de cabeza de tener que usar propiedades como `clear` u `overflow`.

- **Propiedad `visibility`**

Mediante el uso de las propiedades `visibility` es posible ocultar un elemento manteniendo intacto su espacio (al contrario de lo que ocurriría con `display: none`). Con esta propiedad es posible ocultar elementos del diseño sin afectar al resto de elementos.

| | |
|-------------------|---|
| visibility | Visibilidad de un elemento |
| Valores | <code>visible</code> <code>hidden</code> <code>collapse</code> <code>inherit</code> |
| Se aplica a | Todos los elementos |
| Valor inicial | <code>visible</code> |
| Descripción | Permite hacer visibles e invisibles a los elementos |

- **Propiedad `overflow`**

Mediante el uso de esta propiedad es posible controlar el comportamiento de los contenidos cuando éstos se salen de su elemento contenedor. En el diseño mediante el modelo de caja es habitual que el contenido se desborde fuera de su elemento contenedor. Esto ocurre cuando al elemento contenedor se le aplican unas dimensiones concretas mediante el uso de las propiedades `width` y `height`.

| | |
|-----------------|---|
| overflow | Parte sobrante de un elemento |
| Valores | <code>visible</code> <code>hidden</code> <code>scroll</code> <code>auto</code> <code>inherit</code> |
| Se aplica a | Elementos de bloque y celdas de tablas |
| Valor inicial | <code>visible</code> |
| Descripción | Permite controlar los contenidos sobrantes de un elemento |

Los valores de la propiedad `overflow` tienen el siguiente significado:

- ✓ **visible:** el contenido no se corta y se muestra sobresaliendo la zona reservada para visualizar el elemento. Este es el comportamiento por defecto.
- ✓ **hidden:** el contenido sobrante se oculta y sólo se visualiza la parte del contenido que cabe dentro de la zona reservada para el elemento.
- ✓ **scroll:** solamente se visualiza el contenido que cabe dentro de la zona reservada para el elemento, pero también se muestran barras de *scroll* que permiten visualizar el resto del contenido.

- **Propiedad z-index**

Además de posicionar una caja de forma horizontal y vertical, CSS permite controlar la posición tridimensional de las cajas posicionadas. De esta forma, es posible controlar las cajas que se muestran delante o detrás de otras cajas cuando se producen solapamientos.

La posición tridimensional se establece sobre un tercer eje llamado Z y se controla mediante la propiedad z-index. Utilizando esta propiedad es posible crear páginas complejas con varios niveles o capas.

A continuación se muestra la definición formal de la propiedad z-index:

| z-index | Orden tridimensional |
|----------------|--|
| Valores | auto <numero> inherit |
| Se aplica a | Elementos que han sido posicionados explícitamente |
| Valor inicial | auto |
| Descripción | Establece el nivel tridimensional en el que se muestra el elemento |

Cuanto más alto sea el valor numérico, más cerca del usuario se muestra la caja. Un elemento con z-index: 10 se muestra por encima de los elementos con z-index: 8 o z-index: 9, pero por debajo de elementos con z-index: 20 o z-index: 50.

DISEÑOS WEB FIJOS Y FLUIDOS

Diseños web fijos

Un diseño web fijo es aquel que permanece inalterable sea cual sea el dispositivo donde se visualiza el sitio web. Es indiferente al tamaño de la pantalla y a la anchura del navegador web que la visualiza.

Sean cuales sean las características del dispositivo, pantalla o navegador, siempre se visualiza el mismo tamaño preestablecido por el diseñador web en píxeles.

Las características del diseño web fijo son las siguientes:

- Tamaños fijos medidos en píxeles.
- No reaccionan al tamaño de la pantalla.
- Orientado a resoluciones de escritorio.
- No están optimizados para dispositivos móviles.
- Fáciles de crear.
- En resoluciones muy amplias el espacio no es aprovechado, mientras que en resoluciones pequeñas aparecerá una barra de desplazamiento horizontal, lo cual afecta la usabilidad del sitio

Diseños web fluidos

Los diseños web fluidos (también conocidos como “diseños líquidos”) fueron creados para no tener un tamaño definido, y permitir que los monitores pueden condicionar el tamaño de salida.

Usando porcentajes (%) podremos definir el tamaño de los elementos de nuestra página web para definir un diseño fluido que se adapte a cualquier monitor.

Las características de los diseños web fluidos son las siguientes:

- Diseños web adaptables al tamaño del navegador o “viewport”.
- Se construyen definiendo los anchos de los elementos con porcentajes.
- Requiere más planeación y verificación.
- Efectos negativos en pantallas muy pequeñas.

ALTURAS/ANCHURAS MÁXIMAS Y MÍNIMAS

Cuando se diseña la estructura de una página web, se debe tomar la decisión de optar por un diseño de anchura fija o un diseño cuya anchura se adapta a la anchura de la ventana del navegador.

Sin embargo, la mayoría de las veces sería conveniente una solución intermedia: que la anchura de la página sea variable y se adapte a la anchura de la ventana del navegador, pero respetando ciertos límites. En otras palabras, que la anchura de la página no sea tan pequeña como para que no se puedan mostrar correctamente los contenidos y tampoco sea tan ancha como para que las líneas de texto no puedan leerse cómodamente.

CSS define cuatro propiedades que permiten limitar la anchura y altura mínima y máxima de cualquier elemento de la página. Las propiedades son max-width, min-width, max-height y min-height.

| | |
|------------------|---|
| max-width | Anchura máxima |
| Valores | <medida> <porcentaje> none inherit |
| Se aplica a | Todos los elementos salvo filas y grupos de filas de tablas |
| Valor inicial | none |
| Descripción | Permite definir la anchura máxima de un elemento |

| | |
|------------------|---|
| min-width | Anchura mínima |
| Valores | <medida> <porcentaje> inherit |
| Se aplica a | Todos los elementos salvo filas y grupos de filas de tablas |
| Valor inicial | 0 |
| Descripción | Permite definir la anchura mínima de un elemento |

| | |
|-------------------|---|
| max-height | Altura máxima |
| Valores | <medida> <porcentaje> none inherit |
| Se aplica a | Todos los elementos salvo columnas y grupos de columnas de tablas |
| Valor inicial | none |
| Descripción | Permite definir la altura máxima de un elemento |

| | |
|-------------------|---|
| min-height | Altura mínima |
| Valores | <medida> <porcentaje> inherit |
| Se aplica a | Todos los elementos salvo columnas y grupos de columnas de tablas |
| Valor inicial | 0 |
| Descripción | Permite definir la altura mínima de un elemento |

Las propiedades que definen la altura y anchura máxima y mínima se pueden aplicar a cualquier elemento, aunque solamente suelen utilizarse para estructurar la página. En general, las propiedades más utilizadas son max-width y min-width, ya que no es muy habitual definir alturas máximas y mínimas.

De esta forma, para conseguir un diseño de anchura variable pero controlada, se podrían utilizar reglas CSS como la siguiente:

```
#contenedor {
    width: 80%;
    min-width: 500px;
    max-width: 900px;
    margin: 0 auto;
}
```

Con esto estamos consiguiendo que el ancho del elemento con id “contenedor” sea del 80% pero que siempre se mantenga entre los límites de 900px y 500px.

III. Procedimiento

Ejercicio 1. Usando posicionamiento y visualización

1. Crear una carpeta en Mis Documentos con el nombre: **NºCarnet_Guia3**, Nota: Cambiar NºCarnet por su número de carnet de la Universidad.
2. Dentro de la carpeta creada en el paso anterior, crear una carpeta llamada “css”.
3. Abrir en el editor de código de su preferencia la carpeta creada en el paso 1.
4. Crear un nuevo archivo dando click derecho sobre el nombre de la carpeta y seleccionando la opción “**New File**”.
5. Dar clic en la opción “**File>Save**” (o Ctrl+s) para guardar el archivo. Guarde el archivo con el nombre “**oscares.html**”.
6. Escribir la estructura básica de un documento HTML en el archivo creado en el paso anterior.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title></title>
</head>
<body>

</body>
</html>
```

7. Colocar como título de la página (etiqueta <title>) el texto “**Nominaciones mejor película**”.

```
<title>Nominaciones Mejor pelicula</title>
```

8. Descargar los recursos de la guía de la página web de la universidad y colocar la carpeta “images” en su carpeta de trabajo.
9. Digitar el siguiente código HTML dentro del cuerpo (body) de la página web.

```
<header>
<div>
  <h1>Nominaciones Oscars 2016</h1>
  <h2>Mejor pelicula</h2>
</div>
</header>
```

```

<section>
  <article>
    
    <h3>Spotlight</h3>
    <p>En 2002, un equipo de reporteros de investigación del Boston Globe destapó los escándalos de pederastia cometidos durante décadas por curas de Massachussets.</p>
  </article>
  <article>
    
    <h3>El renacido</h3>
    <p>En las profundidades de la América salvaje, el explorador Hugh Glass participa junto a su hijo mestizo Hawk en una expedición de tramperos que recolecta pieles. Glass resulta gravemente herido por el ataque de un oso y es abandonado a su suerte por un traicionero miembro de su equipo, John Fitzgerald.</p>
  </article>
  <article>
    
    <h3>Puente de espías</h3>
    <p>James Donovan, un abogado de Brooklyn se ve inesperadamente involucrado en la Guerra Fría entre su país y la URSS cuando la mismísima CIA le encarga una difícil misión: negociar la liberación de un piloto estadounidense capturado por la Unión Soviética.</p>
  </article>
  <article>
    
    <h3>La habitación</h3>
    <p>Para Jack, un niño de cinco años, la habitación es el mundo entero, el lugar donde nació, donde come, juega y aprende con su madre. Por la noche, mamá lo pone a dormir en el armario, por si viene el viejo Nick.</p>
  </article>
  <article>
    
    <h3>Brooklyn</h3>
    <p>En los años 50, la joven Eilis Lacey decide abandonar Irlanda y viajar a los Estados Unidos, concretamente a Nueva York, donde conoce a un chico del que se enamora.</p>
  </article>
  <article>
    
    <h3>Marte</h3>
    <p>Durante una misión tripulada a Marte, el astronauta Mark Watney es dado por muerto tras una terrible tormenta y abandonado por la tripulación, que pone rumbo de vuelta a la Tierra.</p>
  </article>
</section>
<footer>
  <div>
    Ejemplo de posicionamiento Universidad Don Bosco
  </div>
</footer>

```

- Después de colocado el texto correspondiente, visualice los resultados de la página en el navegador de su preferencia.

11. Crear un nuevo archivo y guardarlo en la carpeta css con el nombre **style.css**
12. Vincule la hoja de estilo creada en el paso anterior con su pagina web agregando el siguiente código dentro del head de su documento HTML.

```
<link rel="stylesheet" href="css/style.css">
```

13. Proceda a escribir las siguientes reglas de estilo básicas dentro de su archivo style.css y visualice los cambios en el navegador de su preferencia.

```
@import url(https://fonts.googleapis.com/css?family=Muli);

*{
    margin: 0;
    padding: 0;
}

body{
    font-family: 'Muli', sans-serif;
    text-align: justify;
}
```

Nótese que con estas reglas de estilo se importa una fuente externa desde google fonts. Además se realiza el “reset css” técnica que permite resetear los márgenes y rellenos que los navegadores aplican por defecto a los elementos HTML. Finalmente, se ocupa la fuente que se importó desde google fonts y se justifica todo el texto de la página.

14. A continuación procederemos a escribir las siguientes reglas que permitirán estilizar el header y el footer de la página. Digite las siguientes reglas de estilo y visualice los resultados.

```
header{
    background: #16A085;
    height: 130px;
}

footer{
    margin-top: 20px;
    background: #16A085;
    height: 50px;
}
```



```

header div, footer div{
  width: 80%;
  margin: 0 auto;
  color: white;
}

h1{
  font-size: 2.3rem;
  margin-bottom: 1rem;
  padding-top: 0.5rem;
}

h2{
  font-size: 2rem;
}

```

Note que dentro del header y del footer se coloca un contenedor div y a ese contenedor se le aplica un ancho del 80% y se “centra” mediante el uso de la declaración “margin: 0 auto”.

15. Ahora procedemos a estilizar la etiqueta section asignándole un ancho del 90% y colocándola al centro.

```

section{
  margin: 0 auto;
  width: 90%;
}

```

16. A continuación se procede a estilizar los elementos article asignándole un alto de 340 pixeles y un ancho de 33.33%.

```

section article{
  font-size: 0.9rem;
  height: 340px;
  width: 33.33%;
}

```

17. Procederemos a estilizar las imágenes que están dentro de los artículos asignando un alto de 180 píxeles y un ancho máximo del 80% con respecto al elemento contenedor. Visualice los resultados en el navegador.

```
section article img{  
  height: 180px;  
  max-width: 80%;  
}
```

18. Consiga que los elementos article floten a la izquierda, haciendo esto debería conseguir que se muestren la información de las películas en tres columnas.
19. Notará que hay un pequeño error en el diseño y es que dado que los elementos que están flotando se salen del flujo de la página, el contenedor section se muestra como que no tuviera contenido y hace que el header y el footer se muestren prácticamente juntos. Para solucionar este error aplique la siguiente regla de estilo al contenedor de los artículos flotantes (es decir, al elemento section) y visualice los resultados en el navegador.

```
overflow: hidden;
```

20. Ahora notará que aunque los article se muestran de forma correcta, la información entre ellos está muy unida, para arreglar esto coloque un margen de 0.5% dentro de los artículos y visualice los resultados en su navegador.
21. Notará que el diseño se ha descuadrado porque hemos agregado un margen de 1% a cada article (0.5% a cada lado). Para solventar esto debemos cambiar el ancho de los article y restarle un 1% al valor actual. El nuevo valor de ancho de los artículos debería ser:

```
width: 32.33%;
```

Colocando esto el estilo debería cuadrar nuevamente mostrando los artículos en tres columnas.

22. Ahora le aplicaremos un borde y un relleno a cada uno de los artículos (selector “section article”). Digite las siguientes reglas de estilo en el lugar correspondiente y visualice los resultados en su navegador.

```
border: 2px dashed #16A085;  
padding: 1%;
```

23. Note que el diseño se ha vuelto a descuadrar puesto que los anchos del borde y del relleno se han agregado a las dimensiones de los artículos. Para solucionar este problema tenemos dos opciones: 1. Recalcular el ancho de los artículos considerando el valor del padding y borde asociado. 2. Usar la propiedad box-sizing con su valor border-box para que el relleno y borde de la caja no se sumen al ancho del artículo sino que sean parte de ese ancho. Seleccionaremos la

opción 2 por ser la más sencilla. Coloque la siguiente declaración dentro de los estilos de los artículos y visualice los cambios en su navegador.

```
box-sizing: border-box;
```

El problema debería haberse resuelto y la sección de su página debería lucir de la siguiente manera:



24. Ahora procederemos a solventar un par de errores. Reduzca el ancho de su navegador y notará que hay un punto en el que el contenido de los artículos y del “header” se desborda de su contenedor. Para solucionar este comportamiento vamos a ocultar el contenido que sobresalga del contenedor agregando la siguiente declaración al header y a los artículos.

```
overflow: hidden;
```

25. Ahora intentaremos centrar las imágenes de cada artículo colocando la siguiente declaración en el selector “section article img”.

```
margin: 5px auto;
```

26. ¿Logro centrar la imagen? ¿No? Esto se debe a que la regla “margin: medida auto;” permite centrar únicamente elementos de bloque y las imágenes son elementos de línea. Para solventar este problema convierta las imágenes a elementos de bloque usando la siguiente declaración:

```
display: block;
```

Visualice los resultados en su navegador y ahora la imagen debería de estar centrada, si no lo consiguió revise cuidadosamente porque ha cometido algún error.

27. Consiga que el título de cada una de las películas se muestre centrado usando la propiedad `text-align`.
28. Ahora conseguiremos que el texto del footer no se muestre tan pegado al borde superior de la página, aplicando la siguiente regla de estilo:

```
footer div{  
  padding-top: 1rem;  
}
```

Usando posicionamiento relativo

29. Ahora seleccionaremos el segundo artículo, le pondremos posicionamiento relativo y los desplazaremos de su posición original. Cambiaremos el color de fondo del artículo colocándole una pequeña transparencia conveniente para ver lo que hay detrás de la caja. Digite las siguientes reglas de estilo y visualice los resultados en su navegador.

```
section article:nth-child(2){  
  background-color: rgba( 70,130,180,0.9);  
  position: relative;  
  left: 100px;  
  top: 50px;  
}
```

30. ¿Esperaba que el artículo se moviera 100 píxeles a la izquierda y 50 píxeles hacia arriba? ¿Es eso lo que obtuvo? Pregúntese porque el artículo se movió hacia abajo y hacia la derecha con respecto a su posición original. Asegúrese de entender este comportamiento antes de seguir con la guía.
31. ¿Qué ha pasado con las otras cajas? ¿Han tomado el espacio que dejó libre el segundo artículo? ¿Por qué?

Usando posicionamiento absoluto

32. Modifique el posicionamiento del segundo artículo de “relativo” a “absoluto” y visualice los cambios en el navegador.
33. ¿A partir de qué posición se ha desplazado el segundo artículo? El artículo se ha desplazado 100 píxeles desde la izquierda y 50 píxeles desde arriba a partir del `body` de la página. Esto ocurre porque el artículo busca el primer antecesor que tenga un posicionamiento no estático (es decir, `relative` u `absolute`) y a partir del origen de ese elemento se calcula su desplazamiento. Al no encontrar ese elemento, el desplazamiento se calcula desde el origen del elemento `body`.

34. Para cambiar ese comportamiento haga los cambios necesarios para que la sección tenga un posicionamiento relativo y visualice los cambios en el navegador.
35. Notará que ahora el desplazamiento se ha tomado desde el origen de la sección y no desde el origen del body.
36. ¿Qué ha pasado con las otras cajas? ¿Han ocupado el espacio que dejó disponible el segundo artículo?
37. A partir del comportamiento visto, explique ¿Cuáles son las diferencias entre el posicionamiento relativo y absoluto?
38. Modifique el posicionamiento del segundo artículo de “absoluto” a “estático” y visualice los cambios en el navegador. Notará que las propiedades de desplazamiento left y top quedan sin efecto dado que un elemento con posicionamiento estático se considera un elemento no posicionado.

Ocultando elementos

39. Consiga que el segundo artículo se oculte utilizando la siguiente declaración:

```
visibility: hidden;
```

Visualice los resultados en su navegador.

40. Elimine la declaración anterior y oculte el segundo artículo esta vez usando la siguiente declaración:

```
display: none;
```

41. A partir de los resultados obtenidos, responda ¿Cuál es la diferencia entre ocultar los elementos ocupando display: none y visibility: hidden?

Ejercicio 2. Construyendo un diseño fijo

1. Creé un archivo HTML llamado "index.html".
2. Guardar el archivo en su carpeta de trabajo.
3. Escribir la estructura básica de un documento HTML en el archivo creado en el paso anterior.
4. Colocar como título de la página el texto "Viajeros, agencia de viajes".
5. Digitar el siguiente código HTML dentro del cuerpo (body) de la página web.

```
<section id="superior">
  <div id="encabezado">
    <nav>
    </nav>
    <div id="imgDesc">
    </div>
  </div>
</section>

<section id="contenido">

</section>
<footer>
  Copyright Lenguajes de marcado y estilo web
</footer>
```

6. Coloque el siguiente código HTML dentro del nav de la página.

```

<ul>
  <li><a href="index.html" id="inicio">Inicio</a></li>
  <li><a href="planes.html" id="planes">Planes</a></li>
  <li><a href="galeria.html" id="galeria">Galeria</a></li>
</ul>
```

7. Coloque el siguiente código dentro del div con id "imgDesc".

```

<div id="descripcion">
  <p>Viajeros es la mejor agencia de viajes de El Salvador. Te ofrecemos
cobertura total a todos los lugares turísticos del país. Tenemos planes
turísticos que se adaptan a tu bolsillo, tiempo y gustos. Deja de perder el
tiempo, y confía tu diversión a los expertos</p>
</div>
```

8. Coloque el siguiente código dentro de la sección con id "contenido".

```
<div class="plan">
  
  <h1>Plan naturaleza</h1>
```

Si te gusta sentir un contacto íntimo con la naturaleza, este plan es para ti. El plan incluye la visita a playas y montañas y caminatas en los lugares más paradisíacos del país.

`<p class="boton">Más información</p>`

`</div>`

`<div class="plan">`

``

`<h1>Plan histórico</h1>`

Si lo tuyo es el turismo histórico, este plan te ofrece un recorrido por las distintas ruinas del país, así como una visita a los principales museos de guerra de El Salvador.

`<p class="boton">Más información</p>`

`</div>`

`<div class="plan">`

``

`<h1>Plan Relax</h1>`

Si quieres recuperarte del estrés de la ciudad este plan podría interesarte. Incluye una visita por los lugares más paradisíacos y tranquilos del país.

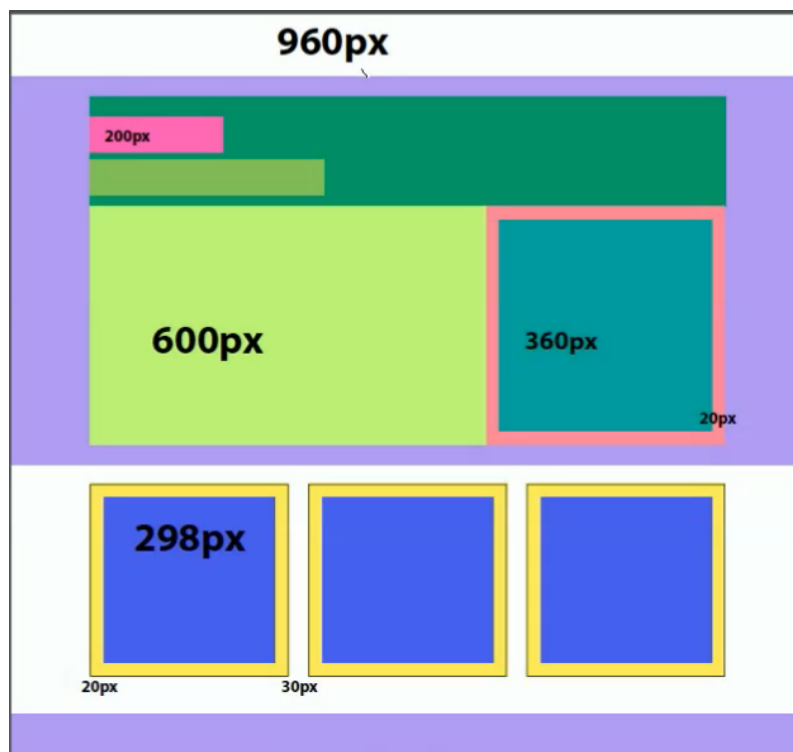
`<p class="boton">Más información</p>`

`</div>`

9. Crear una nueva hoja de estilo, y guardarla dentro de su carpeta css con el nombre “fijo.css”.

10. Vincular su página web con la hoja de estilo creada en el paso anterior.

Nuestro objetivo será aplicar las distintas propiedades CSS estudiadas en el transcurso de la asignatura para crear un layout como el siguiente:



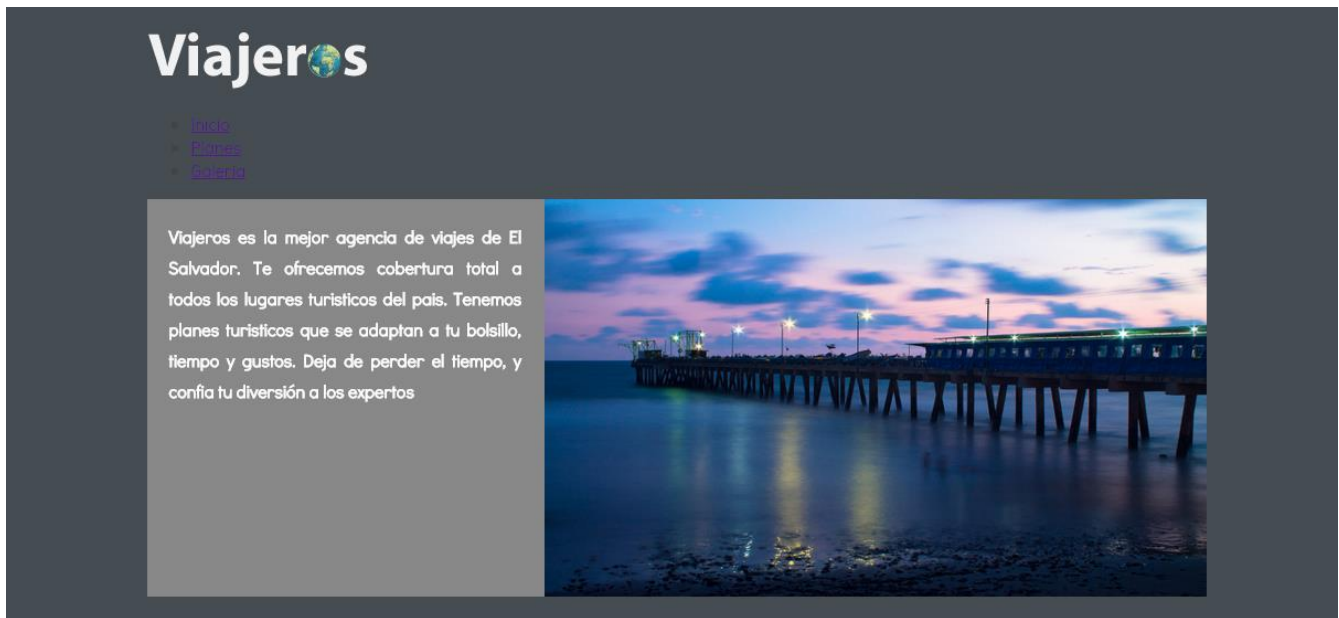
11. Coloque las siguientes reglas de estilo iniciales dentro de su archivo fijo.css

```
/* FUENTES */  
@import url(http://fonts.googleapis.com/css?family=Didact+Gothic);  
  
*{  
    font-family: 'Didact Gothic', sans-serif;  
    margin: 0;  
    padding: 0;  
}  
  
/*Estilos iniciales*/  
body{  
    margin: 0px;  
    background-color: #BBB;  
    color: #444;  
}  
  
h1{  
    font-size: 1.3em;  
    text-align: center;  
    color: #464d52;  
}
```

12. A continuación procedemos a estilizar toda la sección con id “superior”.

```
/* PARTE SUPERIOR */  
#superior{  
    width: 100%;  
    padding: 30px 0px;  
    background-color: #464d52;  
}  
  
#encabezado{  
    margin: 0px auto;  
    width: 960px;  
}  
  
#descripcion{  
    width: 360px;  
    height: 360px;  
    background-color: #888;  
    float: left;  
}  
  
#descripcion p{  
    margin: 20px;  
    color: white;  
    font-weight: bold;  
    line-height: 180%;  
    text-align: justify;  
}
```

13. Hasta el momento su encabezado debería lucir de la siguiente manera:



14. Ahora procederemos a darle estilo al **menú de navegación**, para ello crearemos dentro de la carpeta css una hoja de estilo llamada “**menú.css**”, la vincularemos con nuestra página web y escribiremos las siguientes reglas de estilo:

```
/*  MENU  */

nav{
    width: 960px;
    font-size: 1.4em;
}

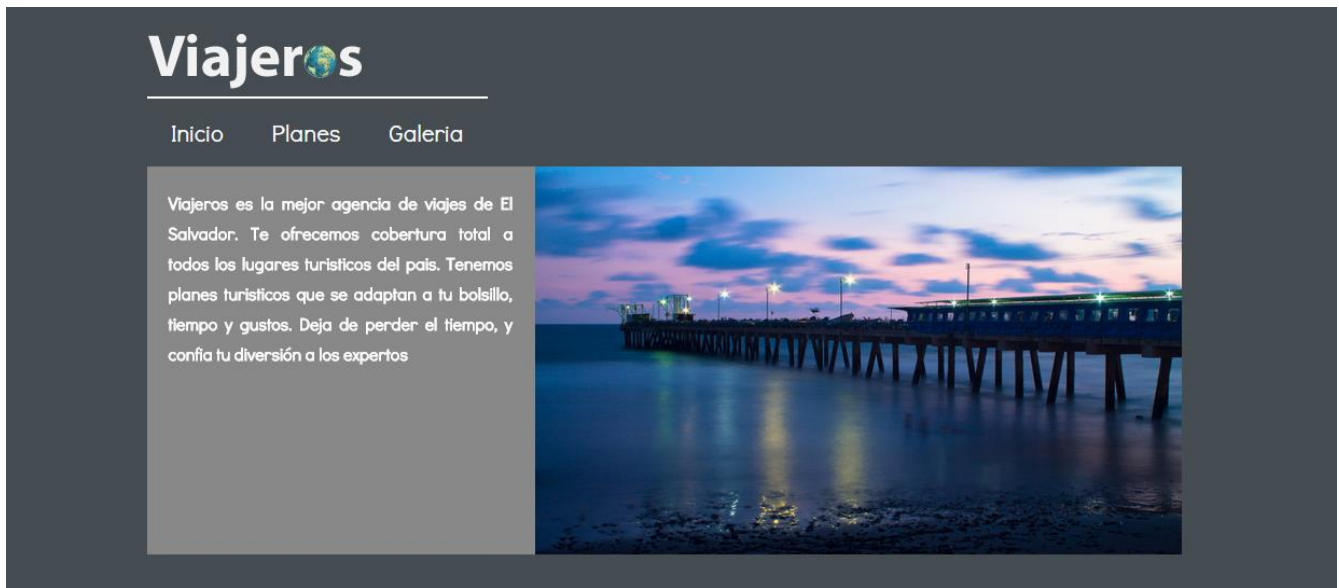
nav ul{
    list-style: none;
    margin: 0px 0px 65px 0px;
}

nav ul li{
    float: left;
    margin: 0px;
    padding: 10px 0px 0px 0px;
    border-top: solid 2px white;
}

nav ul li a {
    text-decoration: none;
    padding: 0.3em 0.5em;
    margin: 0em 0.5em;
    display: block;
    color: white;
}

}
```

15. Hasta el momento su encabezado debería lucir de la siguiente manera:



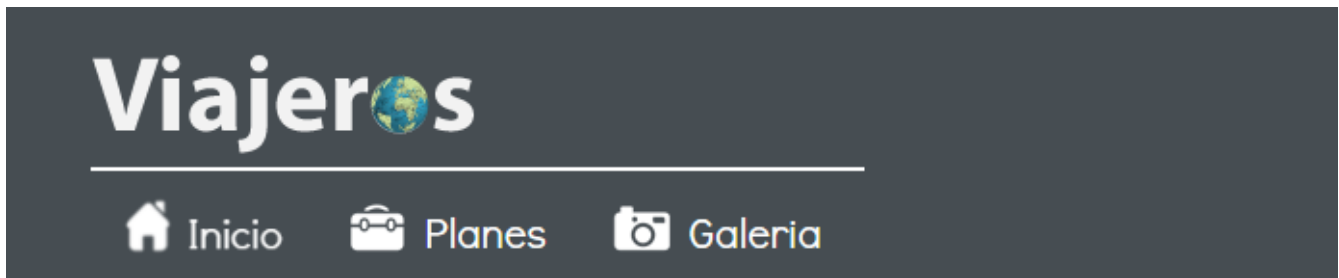
16. Ahora conseguiremos que los vínculos del menú cambien de color y escala cuando el usuario pase el cursor sobre ellos.

```
nav ul li a:hover{  
  color: #AAA;  
  transform:scale(1.25);  
}
```

17. Mediante transiciones consiga que tanto el cambio de color como el cambio de escala se hagan de forma suave y progresiva. La transición del color debe durar 1 segundo y la transición de la escala debe durar 3 segundos.
18. Finalmente agregaremos unos iconos en cada uno de los ítems del menú para mejorar la presentación del mismo.

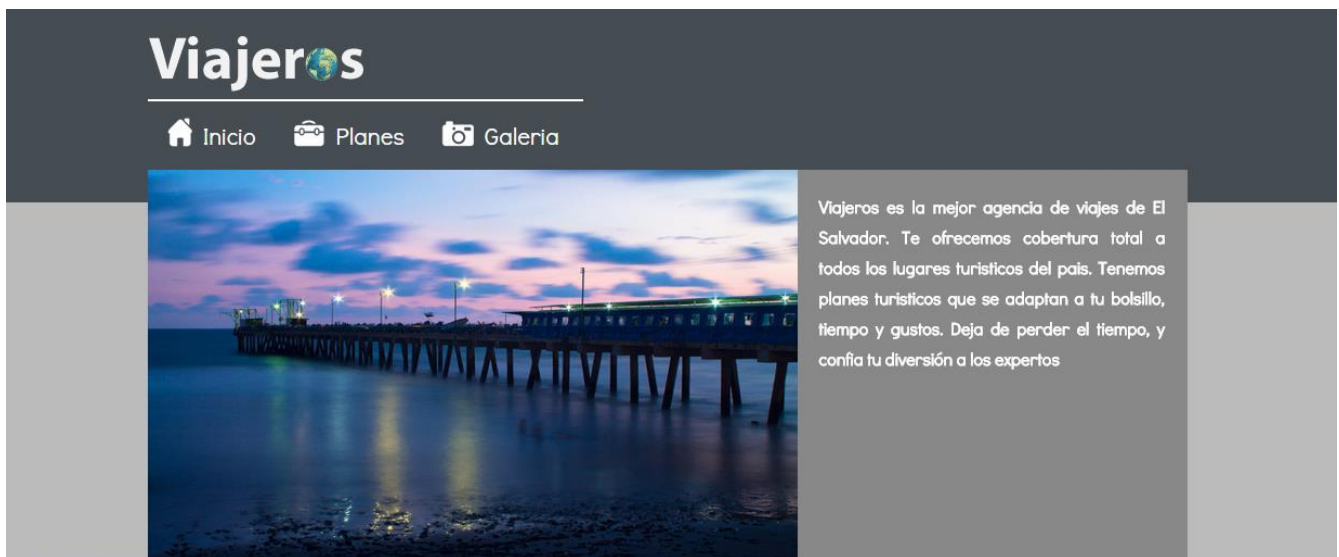
```
nav ul li a#inicio{  
  background: url(../img/turismo/home.png) no-repeat 0px 3px;  
  padding-left: 40px;  
}  
nav ul li a#planes{  
  background: url(../img/turismo/planes.png) no-repeat 0px 3px;  
  padding-left: 40px;  
}  
nav ul li a#galeria{  
  background: url(../img/turismo/galeria.png) no-repeat 0px 3px;  
  padding-left: 40px;  
}
```

19. Si siguió los pasos al pie de la letra las indicaciones, su menú debería lucir de la siguiente manera:



20. Con esto hemos finalizado el menú de navegación. **Cierre el archivo “menú.css” puesto que las siguientes reglas de estilo deberá colocarlas dentro del archivo “fijo.css”.**
21. Ahora finalizaremos el estilo de la sección con id “superior”. Consiga que la imagen que se encuentra dentro del div con id imgDesc flote a la izquierda (estas reglas de estilo deben colocarse en el archivo “fijo.css”)

El resultado obtenido debería ser similar al siguiente:



22. Note que la imagen está flotando a la izquierda, pero parece que se ha salido del contenedor. Este problema tan habitual cuando se usa la propiedad float ya lo hemos estudiado y resuelto en guías pasadas. La solución consiste en aplicar la regla “overflow:hidden” sobre el contenedor del elemento flotado (en este caso el contenedor sería el div con id “imgDesc”) para forzar que el contenedor “envuelva” a los elemento flotantes. Resuelva el error y visualice los resultados en su navegador. Debería obtener un resultado similar al siguiente:



Viajeros es la mejor agencia de viajes de El Salvador. Te ofrecemos cobertura total a todos los lugares turísticos del país. Tenemos planes turísticos que se adaptan a tu bolsillo, tiempo y gustos. Deja de perder el tiempo, y confía tu diversión a los expertos

23. Ahora procederemos a aplicar los estilos básicos para la sección con id “principal” y para el footer de la página.

```
/*Contenido principal de las paginas*/
#contenido{
    width: 960px;
    margin: 0px auto;
    clear: left;
}

/* PIE DE LA PAGINA*/
footer{
    text-align: center;
    margin: 0;
    font-size: .9em;
    line-height: 2;
    clear: both;
    background-color: #464d52;
    color: white;
    width: 100%;
}
```

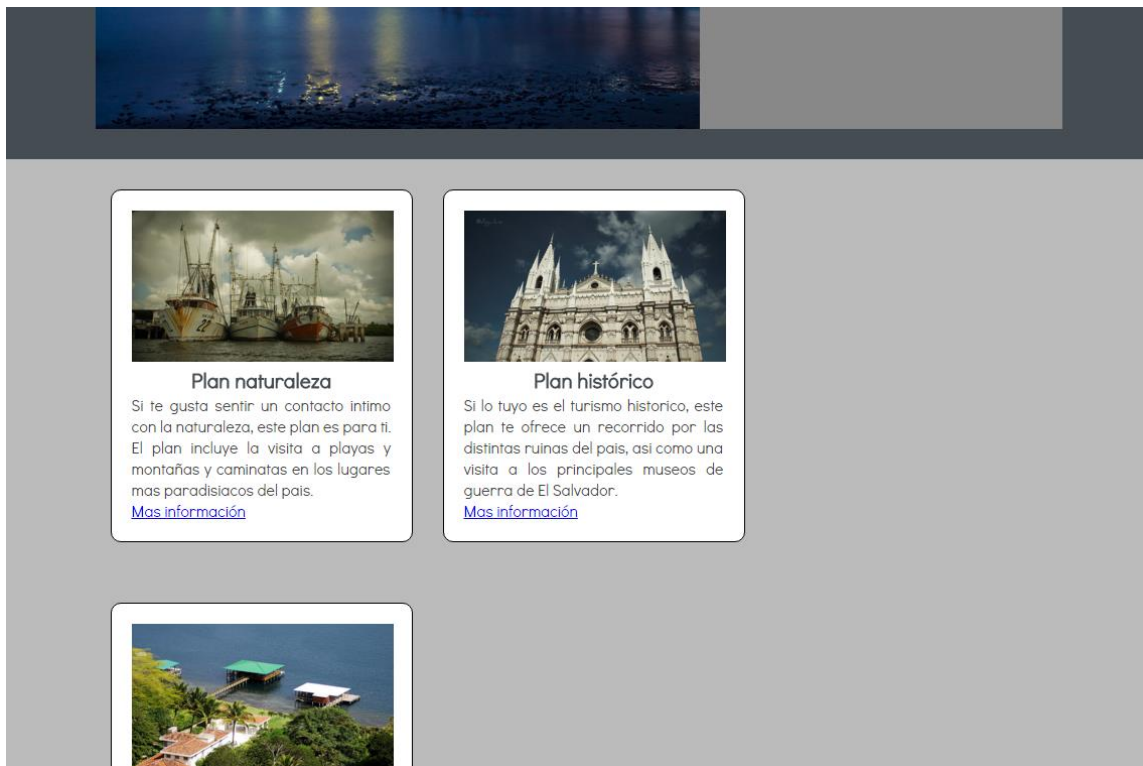
24. Con estos estilos ya tendríamos lista la plantilla que vamos a ocupar en todas las páginas de nuestro sitio web.

25. Ahora escribiremos los estilos particulares para la página index.html. En esta página tenemos algunos div con la clase “plan” los cuales queremos que se muestren en tres columnas.

```
/*  CONTENIDO DE LA PAGINA INDEX */

.plan{
  width: 258px;
  padding: 20px;
  background-color:white;
  margin: 30px 15px;
  border: solid 1px black;
  border-radius: 10px;
  float: left;
  text-align: justify;
}
```

Visualice los resultados en su navegador y notará que aunque los anchos, márgenes y rellenos están colocados tal cual está en el layout presentado al inicio del ejercicio, la tercera columna no cabe y se va hacia abajo.



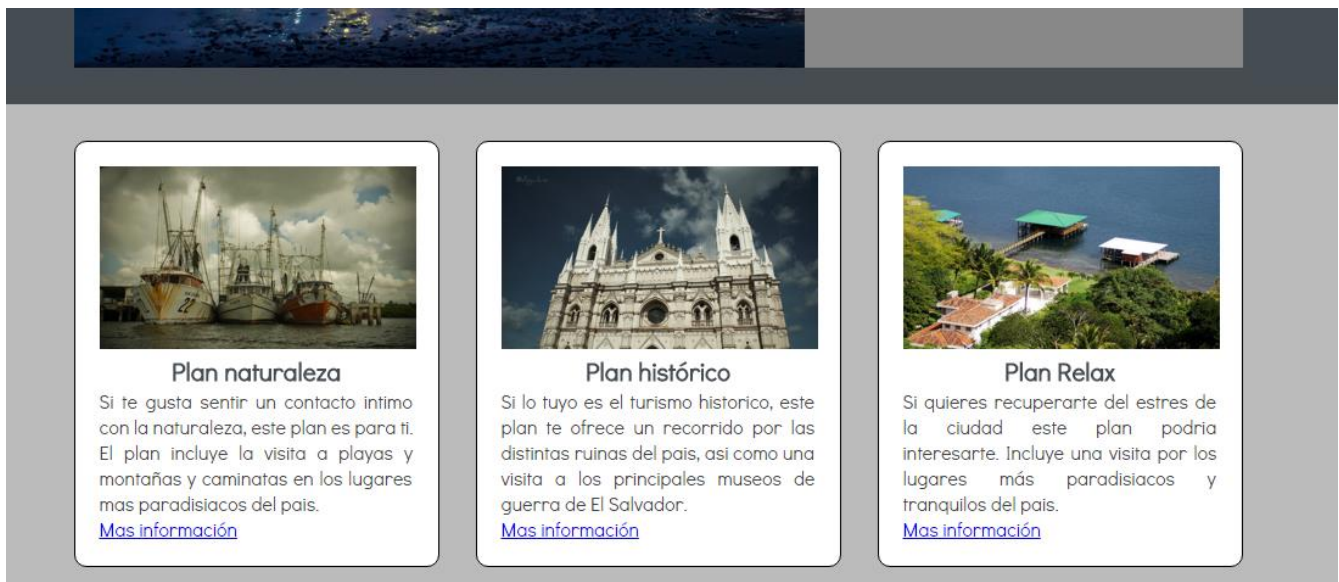
El problema se debe a que el primer div tiene un margen a la izquierda el cual no debería estar. Lo mismo sucede con el tercer div que tiene un margen innecesario a la derecha.

26. Elimine estos márgenes innecesarios mediante las siguientes reglas de estilo:

```
/* Primer div, cuarto div, .... */
#contenido div:nth-child(3n+1){
    margin-left: 0px;
}

/* Tercer div, sexto div, etc */
#contenido div:nth-child(3n){
    margin-right: 0px;
}
```

El resultado obtenido debería ser similar al siguiente:



27. Ahora haremos que los enlaces que los enlaces que tienen la clase botón, tengan dicha apariencia. Para ello, digite las siguientes reglas de estilo.

```
.boton{
    background-color:#464d52;
    padding: 7px;
    border-radius: 5px;
    color: white;
    margin-top: 10px;
    font-size: .9em;
    text-align: center;
    vertical-align: middle;
    cursor: pointer;
    transition: background 1s;
}
```



```

.boton:hover{
    background: #5077c7;
}

.boton a{
    text-decoration: none;
    color: white;
    width: 100%;
    display: block;
}

```

Página planes.html

Para construir las restantes páginas del sitio, la pagina index.html nos servirá como plantilla.

28. Cree una copia de su archivo index.html y guárdela con el nombre planes.html.
29. Dentro del archivo **planes.html** elimine todo el contenido de la sección con id “contenido” y reemplácelo por el siguiente código HTML.

```

<div class="planes" id="galeriaNaturaleza">
    <h1>Plan naturaleza</h1>
    
    <p>Si te gusta sentir un contacto íntimo con la naturaleza, este
plan es para ti. El plan incluye la visita a playas y montañas y caminatas en
los lugares más paradisíacos del país.</p>
    <p class="precio">5 días y 4 noches por $1000</p>
</div>
<div class="planes" id="galeriaHistorico">
    <h1>Plan histórico</h1>
    
    <p>Si lo tuyo es el turismo histórico, este plan te ofrece un
recorrido por las distintas ruinas del país, así como una visita a los
principales museos de guerra de El Salvador.</p>
    <p class="precio">5 días y 4 noches por $1000</p>
</div>
<div class="planes" id="galeriaRelax">
    <h1>Plan Relax</h1>
    
    <p>Si quieres recuperarte del estrés de la ciudad este plan
podría interesarte. Incluye una visita por los lugares más paradisíacos y
tranquilos del país.</p>
    <p class="precio">5 días y 4 noches por $1000</p>
</div>

```

30. Añada las siguientes reglas de estilo dentro de su archivo fijo.css

```
/* CONTENIDO DE LA PAGINA PLANES */

.planes{
  width: 900px;
  background: white;
  padding: 10px 30px;
}

.planes h1{
  text-align: left;
  font-size: 1.5em;
}

.planes .precio{
  font-size: 1.1em;
  margin: 20px 0px 5px;
  padding-bottom: 15px;
  border-bottom: solid 2px grey;
  font-weight: bold;
}
```

El contenido de su página planes.html debería lucir de la siguiente manera:

Plan naturaleza



Si te gusta sentir un contacto íntimo con la naturaleza, este plan es para ti. El plan incluye la visita a playas y montañas y caminatas en los lugares más paradisíacos del país.

5 días y 4 noches por \$1000

Página galeria.html

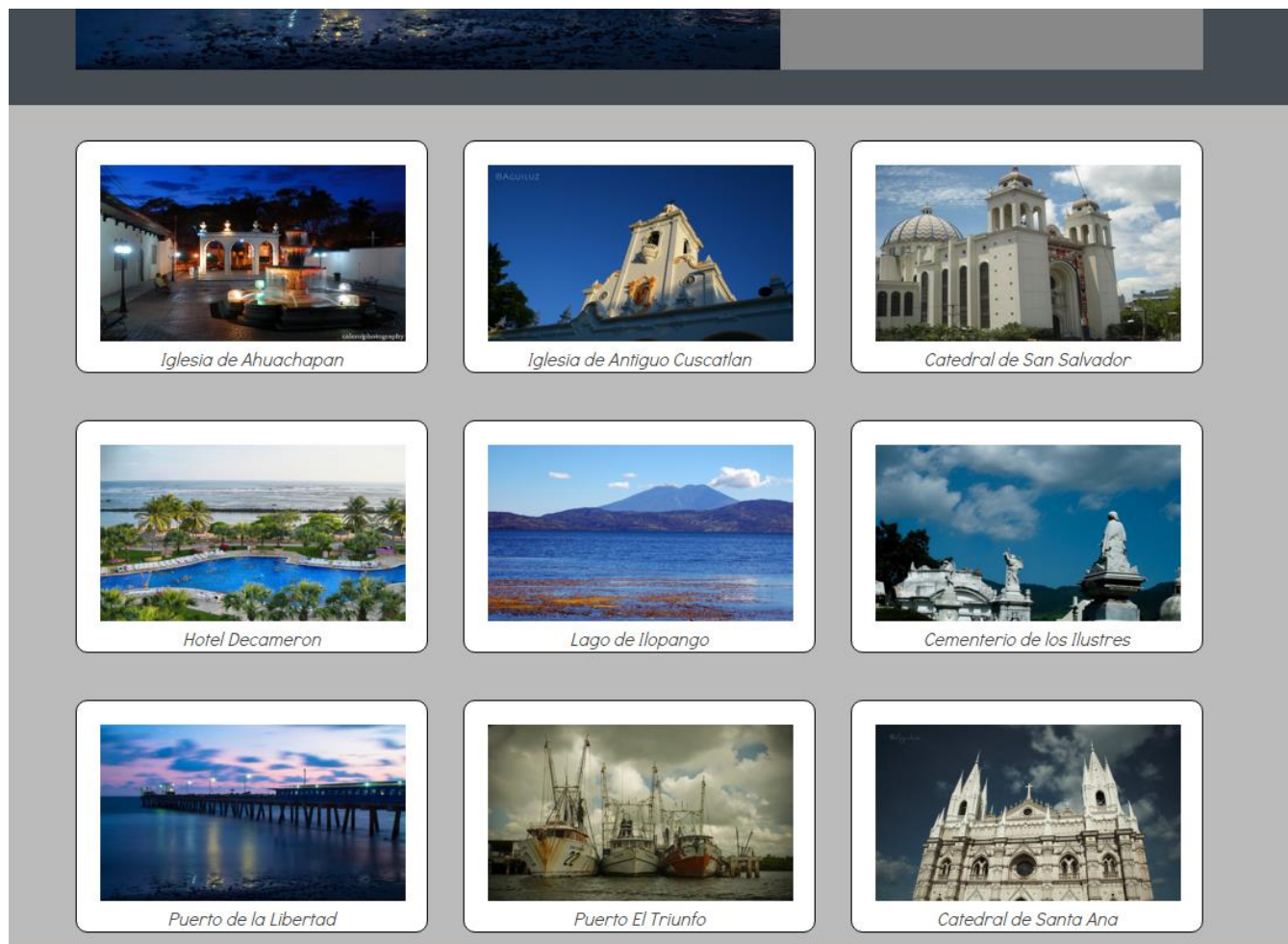
31. Cree una copia de su archivo index.html y guárdela con el nombre galeria.html.
32. Dentro del archivo **galeria.html** elimine todo el contenido de la sección con id "contenido" y reemplácelo por el siguiente código HTML:

```
<div id="galeria">
    <div class="plan">
        
        <p>Iglesia de Ahuachapan</p>
    </div>
    <div class="plan">
        
        <p>Iglesia de Antiguo Cuscatlan</p>
    </div>
    <div class="plan">
        
        <p>Catedral de San Salvador</p>
    </div>
    <div class="plan">
        
        <p>Hotel Decameron</p>
    </div>
    <div class="plan">
        
        <p>Lago de Ilocpango</p>
    </div>
    <div class="plan">
        
        <p>Cementerio de los Ilustres</p>
    </div>
    <div class="plan">
        
        <p>Puerto de la Libertad</p>
    </div>
    <div class="plan">
        
        <p>Puerto El Triunfo</p>
    </div>
    <div class="plan">
        
        <p>Catedral de Santa Ana</p>
    </div>
</div>
```

33. Añada las siguientes reglas de estilo dentro de su archivo fijo.css

```
/* CONTENIDO DE LA PAGINA GALERIA*/  
#galeria .plan p{  
    text-align: center;  
    font-style: italic;  
}  
  
#galeria .plan{  
    font-size: 1em;  
    margin-bottom: 10px;  
    padding-bottom: 1px;  
}
```

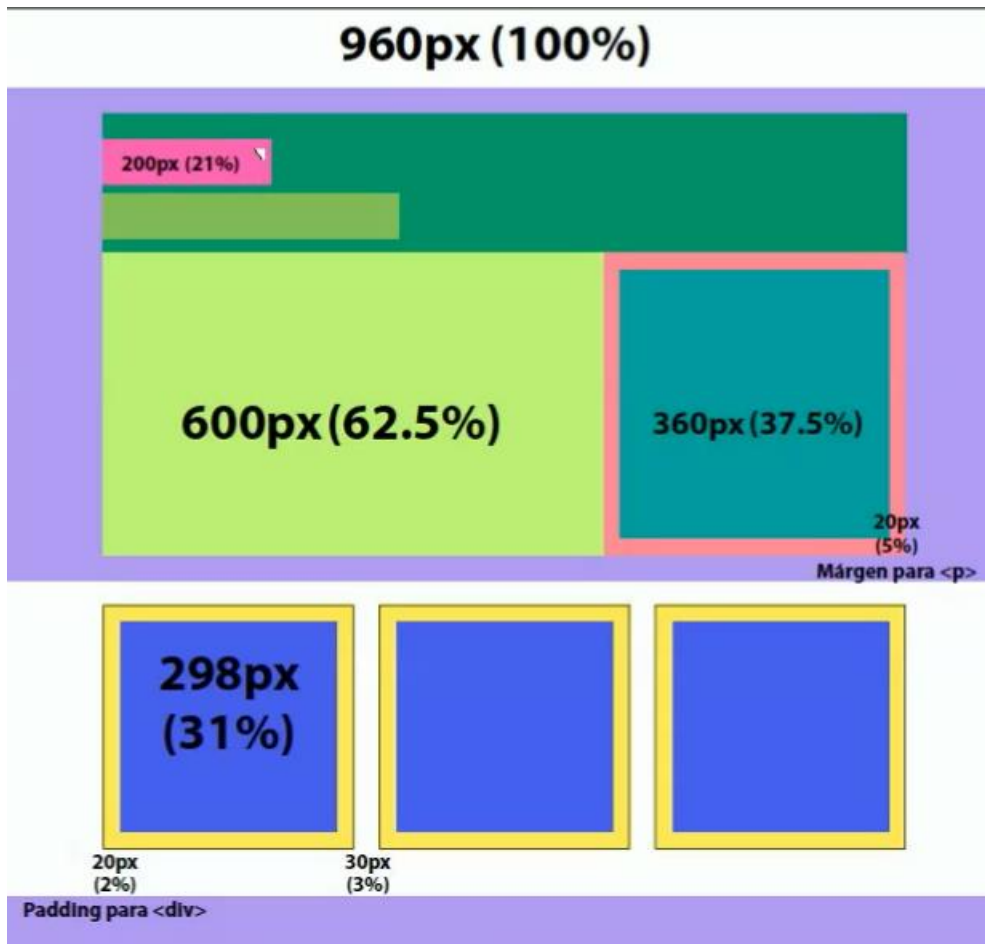
El contenido de su página galeria.html debería lucir de la siguiente manera:



Ejercicio 3. Construyendo un diseño líquido

En este ejercicio, vamos a modificar las hojas de estilos ocupadas en el ejercicio anterior para que el sitio web presente un diseño líquido, fluido o flexible, es decir, que se adapte a los cambios en el ancho del viewport.

El objetivo es que el diseño del sitio web se acople al siguiente layout:



1. Cree una copia de su hoja de estilo fijo.css y guárdela con el nombre flexible.css
2. Cree una copia de su hoja de estilo menú.css y guárdela con el nombre menú_flexible.css
3. Vincule las hojas de estilo flexible.css y menú_flexible.css a sus páginas web (las paginas ya NO usaran las hojas de estilo fijo.css y menú.css).
4. Dentro de su hoja de estilo menú_flexible.css cambie el ancho del nav para que ahora cubra el 100% del ancho disponible. Su regla de estilo debería lucir de la siguiente manera:

```
nav{  
  width: 100%;  
  font-size: 1.4em;  
}
```

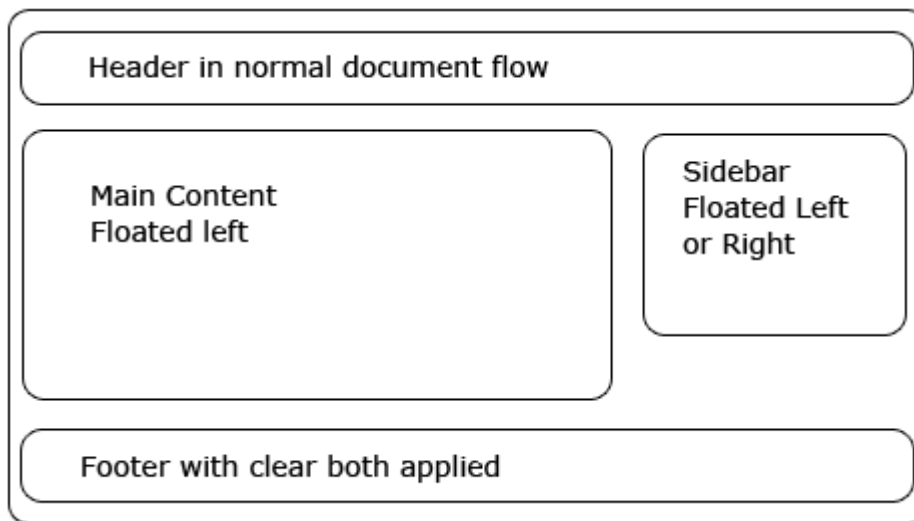
5. En la siguiente tabla se muestran los cambios que deben hacerse en la hoja de estilo flexible.css para conseguir un diseño líquido (Solamente debe modificar los propiedades que salen acá, las otras déjelas tal y como están).

| Selector | Valor original | Valor a modificar |
|---------------------------------------|---|--|
| #superior | padding: 30px 0px; | padding: 2% 0px; |
| #encabezado | width: 960px; | width: 80%; min-width: 650px;/* YA NO REACCIONARA SI LA CAJA ES MAS PEQUEÑA DE 650px*/ max-width: 960px;/*YA NO REACCIONARA SI LA CAJA ES MAS GRANDE QUE 960 px*/ |
| #descripcion | width: 360px; height: 360px; | width: 37.5%; /* 360/960=0.375*/ height: auto; |
| #descripcion p | margin: 20px; | margin: 5%; /* 20/360 =5% */ |
| #superior #encabezado #logo | ESTE SELECTOR NO ESTA, DEBE AGREGARLO | margin: 2% 0px; /*23/960=2%*/ width: 21%; /*200/960= 21%*/ |
| #superior #encabezado #imgDesc img | float: left; | float: left; width: 62.5%; /* 600/960= 62.5% */ height: auto; |
| #contenido | width: 960px; | width: 80%; margin: 0px auto; min-width: 650px;/* YA NO REACCIONARA SI LA CAJA ES MAS PEQUEÑA DE 650px*/ max-width: 960px;/*YA NO REACCIONARA SI LA CAJA ES MAS GRANDE QUE 960 px*/ |
| .plan | width: 258px; padding: 20px; margin: 30px 15px; | width: 26.7%; /* 258/960= 27%*/ padding: 2%; /* 20/960=2%*/ margin: 30px 1.5%; |
| .plan img | ESTE SELECTOR NO ESTA, DEBE AGREGARLO | width: 100%; height: auto; |
| .planes | width: 900px; padding: 10px 30px; | width: 94%; padding: 1% 3%; |
| .planes img | ESTE SELECTOR NO ESTA, DEBE AGREGARLO | width: 100%; border-radius:15px; |
| #galeria .plan | margin-bottom: 10px; | margin: 30px 1.4% 10px; display: inline-block; float: none; vertical-align: top; |

6. Verifique que todo el contenido de la página fluya correctamente y corrija cualquier error detectado.

IV. Ejercicios complementarios

1. Modifique el ejemplo 1 de la guía de forma que el header y footer de la página tengan posicionamiento fijo, es decir se queden “pegados” en la parte superior e inferior de la pantalla del navegador aun cuando se haga “scroll” en la página. Considere que debe conseguir que el contenido de la sección no se quede oculto debajo de las cajas fijas en ningún momento.
2. Consiga que la imagen que se muestra en la parte superior de su diseño fluido del ejercicio 3, se sustituya por un slider de imágenes manual o automático (las imágenes que debe mostrar en el slider son: puerto_la_libertad.jpg, santa_ana.jpg y salvador_del_mundo.jpg. Estas imágenes están dentro de los recursos de la guía).
3. Construir la página principal de un blog sobre un tema de su elección considerando el siguiente layout como base:



Debe procurar obtener el resultado más estético posible y puede usar las propiedades de posicionamiento y visualización que considera necesarias. Además puede usar cualquiera de las propiedades vistas durante el transcurso de la materia e incluso investigar propiedades y técnicas adicionales para obtener un resultado de calidad.

El blog debe tener el resumen de 4 entradas como mínimo y deben usarse imágenes, vínculos y cualquier otro elemento que aporte a la presentación de la página.