



ESCUELA SUPERIOR DE ECONOMÍA Y NEGOCIOS INGENIERÍA DE SOFTWARE Y NEGOCIOS DIGITALES

CICLO 03-2025

GUIA DE LABORATORIO N° 3

Nombre de la practica: Posicionamiento, visualización y maquetación de sitios web

Tiempo estimado: 2 horas

Materia: Desarrollo Web I

I. Objetivos

Que el estudiante sea capaz de:

1. Utilizar los diferentes tipos de posicionamiento CSS y diferenciarlos apropiadamente.
2. Utilizar las propiedades CSS relacionada con la visualización de elementos.
3. Maquetar sitios web con diseños fijos y fluidos

II. Introducción Teórica

Para crear una buena composición de los elementos en pantalla, es necesario conocer todas las opciones posibles al posicionamiento de elementos y, sobre todo, tener algunas nociones de los tipos de elementos con los que trabaja CSS a la hora de posicionarlos en pantalla.

Para cumplir con el box model los navegadores crean una caja para representar cada elemento de una página HTML. Para hacerlo, los navegadores llevan a cabo una compleja operación en la que se tienen en cuenta los siguientes parámetros:

1. Las propiedades width (ancho) y height (alto) de las cajas, en caso de haber especificado unas dimensiones específicas.
2. El tipo de cada elemento HTML, bien si es un elemento de bloque o un elemento de línea.
3. Posicionamiento de las cajas (estático, relativo, absoluto, fijo o flotante).
4. El parentesco y descendencia de cada elemento.
5. Las dimensiones de la ventana del navegador en función de la resolución de pantalla o las dimensiones de las imágenes, entre otros.

A este procedimiento se le conoce comúnmente como “*modelo de formato visual*”, que definido para entenderlo rápidamente puede ser algo así como el modelo que siguen los navegadores para procesar el contenido de un documento para los medios visuales.

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</code> <code>relative</code> <code>absolute</code> <code>fixed</code> <code>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	<code><medida></code> <code><porcentaje></code> <code>auto</code> <code>inherit</code>
Se aplica a	Todos los elementos posicionados
Valor inicial	<code>auto</code>
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	<code>inline</code> <code>block</code> <code>none</code> <code>list-item</code> <code>run-in</code> <code>inline-block</code> <code>table</code> <code>inline-table</code> <code>table-row-group</code> <code>table-header-group</code> <code>table-footer-group</code> <code>table-row</code> <code>table-column-group</code> <code>table-column</code> <code>table-cell</code> <code>table-caption</code> <code>inherit</code>
Se aplica a	Todos los elementos
Valor inicial	<code>inline</code>
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.

III. Procedimiento

Ejercicio 1. Usando posicionamiento y visualización

1. Crear una carpeta 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 su preferencia la carpeta creada en el paso 1 mediante la opción "**File>Open Folder**" del menú ubicado en la parte superior de la ventana.
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 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>
</section>
```

```

<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
  </div>
</footer>

```

10. 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 pixeles 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 (puede considerar hacer esto mediante el uso de flexbox).
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, relativo 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. Notará que ahora el desplazamiento se ha tomado desde el origen de la sección y no desde el origen del body.
35. ¿Qué ha pasado con las otras cajas? ¿Han ocupado el espacio que dejó disponible el segundo artículo?
36. A partir del comportamiento visto, explique ¿Cuáles son las diferencias entre el posicionamiento relativo y absoluto?
37. 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

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

```
visibility: hidden;
```

Visualice los resultados en su navegador.

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

```
display: none;
```

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

Ejercicio 2. Diseñando un menú horizontal multinivel

1. Cree un nuevo archivo y guárdelo en la carpeta creada con su carnet haciendo uso de Ctrl+s o dando click en file, save y colocarle como nombre **menu_horizontal.html**
2. Digite la estructura básica de un documento HTML.

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

</body>
</html>
```

3. Colocar como título de la página (etiqueta **<title>**) el texto **"Menú horizontal"**.
4. Digitar el siguiente código HTML dentro del cuerpo (body) de la página web.

```
<nav>
  <ul>
    <li><a href="#">Guatemala</a></li>
    <li><a href="#">El Salvador</a></li>
    <li><a href="#">Honduras</a></li>
    <li><a href="#">Nicaragua</a></li>
    <li><a href="#">Costa Rica</a></li>
    <li><a href="#">Panamá</a></li>
    <li><a href="#">Belice</a></li>
  </ul>
</nav>
```

5. Crear un nuevo archivo y guardarlo en la carpeta css con el nombre **menu.css**
6. Vincular la hoja de estilo con el archivo HTML creado.
7. Comenzaremos aplicando algunos estilos básicos a la página web

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

body{
  font-family: Arial;
}
```

8. En este ejemplo construiremos un menú horizontal de dos niveles por tanto dividiremos los estilos en tres grupos: Los estilos generales (que se aplicaran a todos los elementos del menú independientemente del nivel), los estilos del primer nivel y los estilos del segundo nivel.

Estilos generales

9. Comenzaremos escribiendo los estilos generales de los elementos del menú. Iniciaremos quitándole las viñetas a cada uno de los elementos de la lista.

```
/* Estilos generales*/  
  
nav ul{  
    list-style-type: none;  
}
```

10. A continuación aumentaremos el alto de la línea de cada elemento de la lista. Digite las siguientes reglas de estilo y visualice los resultados en el navegador:

```
nav li{  
    line-height: 2.5em;  
}
```

11. Ahora se procederá a estilizar cada uno de los vínculos que forman el menú. Digite las siguientes reglas de estilo y visualice los resultados en el navegador.

```
nav a{  
    color: #ddd;  
    display: block;  
    padding: 0 2.5em;  
    text-decoration: none;  
}  
nav a:hover{  
    background: rgba(0,0,0,0.5);  
}
```

Note que en las reglas de estilo anteriores se le ha quitado el subrayado a los enlaces, además se le ha agregado un relleno horizontal para que el enlace se muestre más ancho. Además se le ha aplicado estilo a la pseudo-clase hover para que cambie el color de fondo del vínculo cuando el usuario pase el cursor sobre él.

Estilos del primer nivel

12. Comenzaremos colocando la lista del primer nivel a ancho completo y con un color de fondo. Digite las siguientes reglas de estilo y visualice los cambios en el navegador.

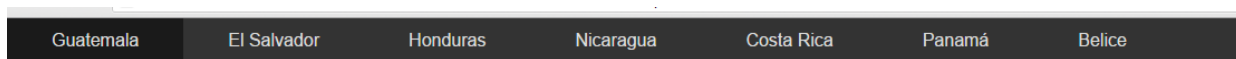
```
/* Primer nivel*/  
  
nav>ul{  
    background-color: #333;  
    width: 100%;  
}
```

13. A continuación conseguiremos que cada uno de los elementos del primer nivel del menú floten a la izquierda. Digite la siguiente regla de estilo y visualice los cambios en el navegador.

```
nav>ul>li{  
    float: left;  
}
```

14. Al visualizar los resultados se habrá dado cuenta que se perdió el color de fondo de la lista, esto es porque la etiqueta ul se quedó sin elementos(es decir, sin altura) dado que los elementos li se “salieron del flujo” al momento en que se pusieron a flotar. Para solucionar este problema debe aplicar la declaración “**overflow:hidden**” al contenedor de la lista (selector **nav>ul**), con esto obligará a que el elemento ul “cubra” a los elementos li.

15. Hasta este punto su menú debe lucir de la siguiente manera:



Segundo nivel

16. Ahora procederemos a realizar el segundo nivel del menú, para ello “anidaremos” una lista, es decir, colocaremos una lista como elemento de otra lista. Para este ejemplo, listaremos los departamentos de El Salvador. Para ello, modificaremos el segundo elemento de la lista original, el código HTML debería lucir de la siguiente manera:


```

<ul>
  <li><a href="#">Guatemala</a></li>
  <li><a href="#">El Salvador</a>
    <ul>
      <li><a href="#">Ahuachapán</a></li>
      <li><a href="#">Santa Ana</a></li>
      <li><a href="#">Sonsonate</a></li>
      <li><a href="#">Usulután</a></li>
      <li><a href="#">San Miguel</a></li>
      <li><a href="#">Morazán</a></li>
      <li><a href="#">La Unión</a></li>
      <li><a href="#">La libertad</a></li>
      <li><a href="#">Chalatenango</a></li>
      <li><a href="#">Cuscatlán</a></li>
      <li><a href="#">San Salvador</a></li>
      <li><a href="#">La Paz</a></li>
      <li><a href="#">Cabañas</a></li>
      <li><a href="#">San Vicente</a></li>
    </ul>
  </li>
  <li><a href="#">Honduras</a></li>
  <li><a href="#">Nicaragua</a></li>
  <li><a href="#">Costa Rica</a></li>
  <li><a href="#">Panamá</a></li>
  <li><a href="#">Belice</a></li>
</ul>

```

17. Hasta este punto la página web debería lucir de la siguiente manera:

Guatemala	El Salvador	Honduras	Nicaragua	Costa Rica	Panamá	Belice
	Ahuachapán					
	Santa Ana					
	Sonsonate					
	Usulután					
	San Miguel					
	Morazán					
	La Unión					
	La libertad					
	Chalatenango					
	Cuscatlán					
	San Salvador					
	La Paz					
	Cabañas					
	San Vicente					

18. El error que ocurre es que los elementos del segundo nivel del menú (es decir, los departamentos) “empujan” a la etiqueta ul principal lo que hace que ese ul cubra a ambos niveles del menú. Para solucionar esto posicionaremos de forma absoluta a la lista del segundo nivel del menú para que se salgan del flujo.

```
nav>ul>li>ul{  
    position: absolute;  
}
```

19. Al visualizar los resultados notará que los elementos de la lista más interior son sacados del flujo pero aún se ven. Nuestro objetivo es ocultar dicha lista hasta que el usuario pase el cursor por el elemento del primer nivel del menú. Para ocultar la lista interior digite la declaración **display:none** dentro del selector de la regla del paso 18 (selector nav>ul>li>ul).
20. Ahora haremos que esa lista interior se muestre cuando se haga “hover” sobre los elementos del primer nivel del menú. Digite la siguiente regla de estilo y visualice los resultados en el navegador:

```
nav>ul>li:hover ul{  
    display: block;  
}
```

21. Al visualizar los resultados notará que el resultado casi está listo y solo nos resta darle un color de fondo sólido a los elementos del segundo nivel de la lista.

```
nav>ul>li>ul>li{  
    background-color: #222;  
}
```

22. Si ha seguido los pasos al pie de la letra debería obtener un resultado como el siguiente:

Guatemala	El Salvador	Honduras	Nicaragua	Costa Rica	Panamá	Belice
	Ahuachapán					
	Santa Ana					
	Sonsonate					
	Usulután					
	San Miguel					
	Morazán					
	La Unión					
	La libertad					
	Chalatenango					
	Cuscatlán					
	San Salvador					
	La Paz					
	Cabañas					
	San Vicente					

Ejercicio 3. 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
</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".

```

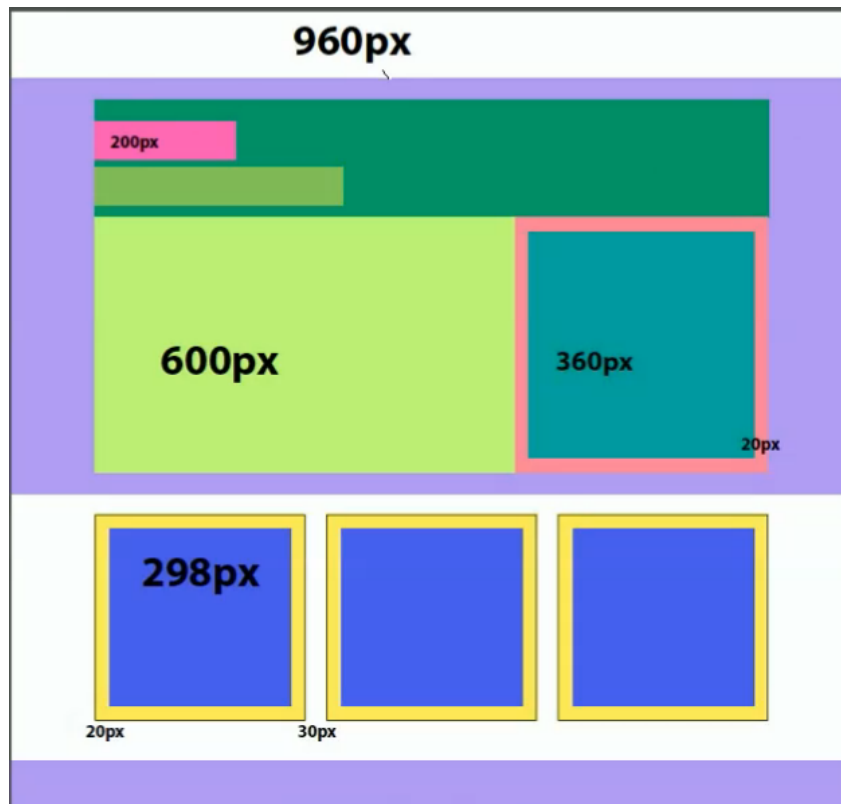
<div id="descripcion">
  <p>Viajeros es la mejor agencia de viajes de El Salvador. Te ofrecemos
cobertura total a todos los lugares turísticos del país. Tenemos planes
turísticos que se adaptan a tu bolsillo, tiempo y gustos. Deja de perder el
tiempo, y confía tu diversión a los expertos</p>
</div>
```

- 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"><a href="planes.html#galeriaNaturaleza">Más
información</a></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"><a href="planes.html#galeriaHistorico">Más
información</a></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"><a href="planes.html#galeriaRelax">Más
información</a></p>
</div>
```

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

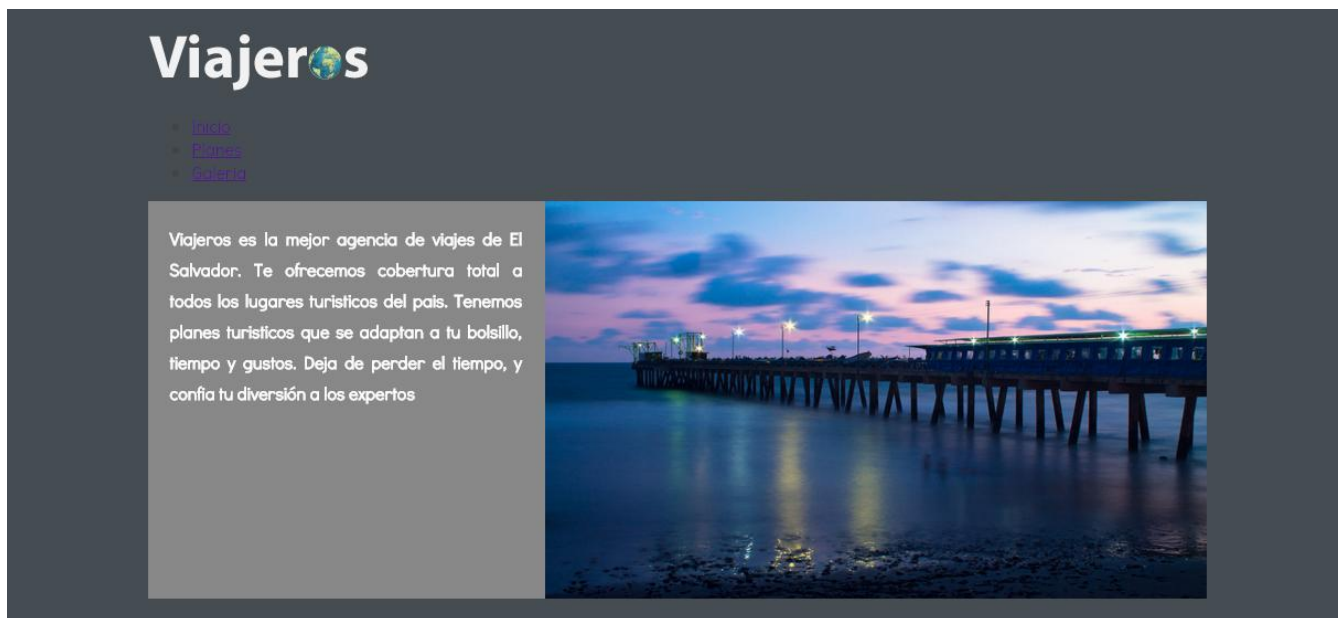
```
/* PARTE SUPERIOR */
#superior{
    width: 100%;
    padding: 30px 0px;
    background-color: #464d52;
}

#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;
}

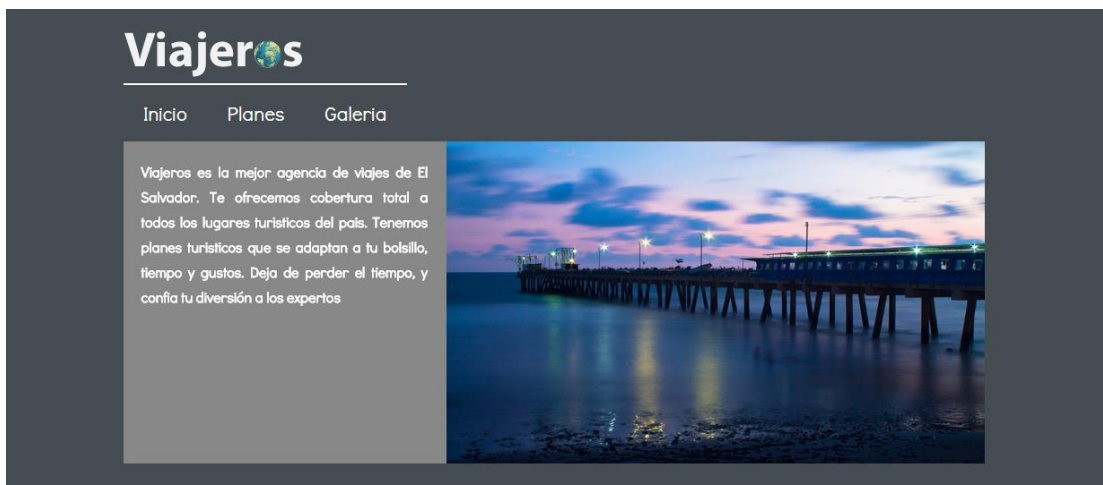
nav ul{
    list-style: none;
    margin: 0px 0px 65px 0px;
}

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:



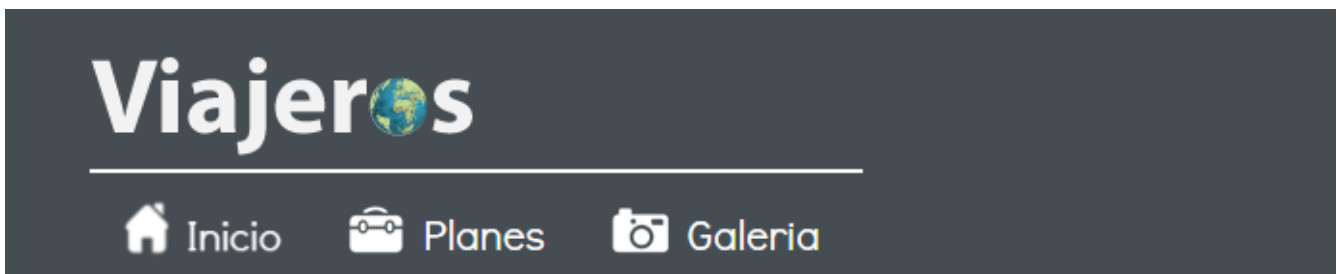
16. Ahora conseguiremos que los vínculos del menú cambien de color y escala cuando el usuario pase el cursor sobre ellos.

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

17. Finalmente agregaremos unos iconos en cada uno de los ítems del menú para mejorar la presentación del mismo.

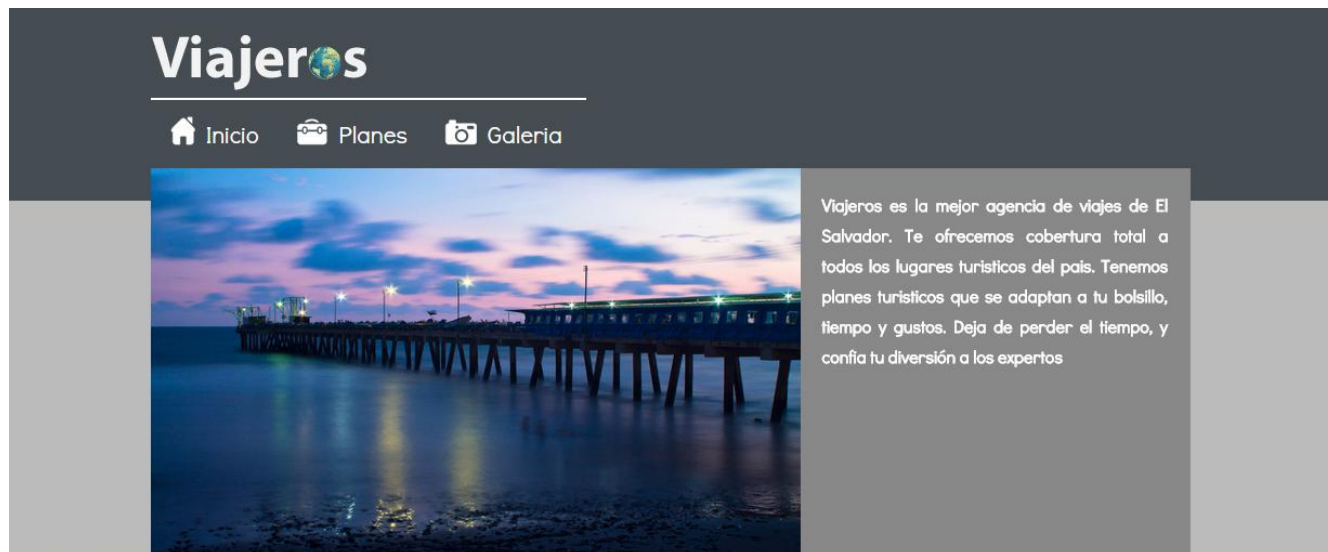
```
nav ul li a#inicio{  
    background: url(../img/turismo/home.png) no-repeat 0px 3px;  
    padding-left: 40px;  
}  
nav ul li a#planes{  
    background: url(../img/turismo/planes.png) no-repeat 0px 3px;  
    padding-left: 40px;  
}  
  
nav ul li a#galeria{  
    background: url(../img/turismo/galeria.png)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

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

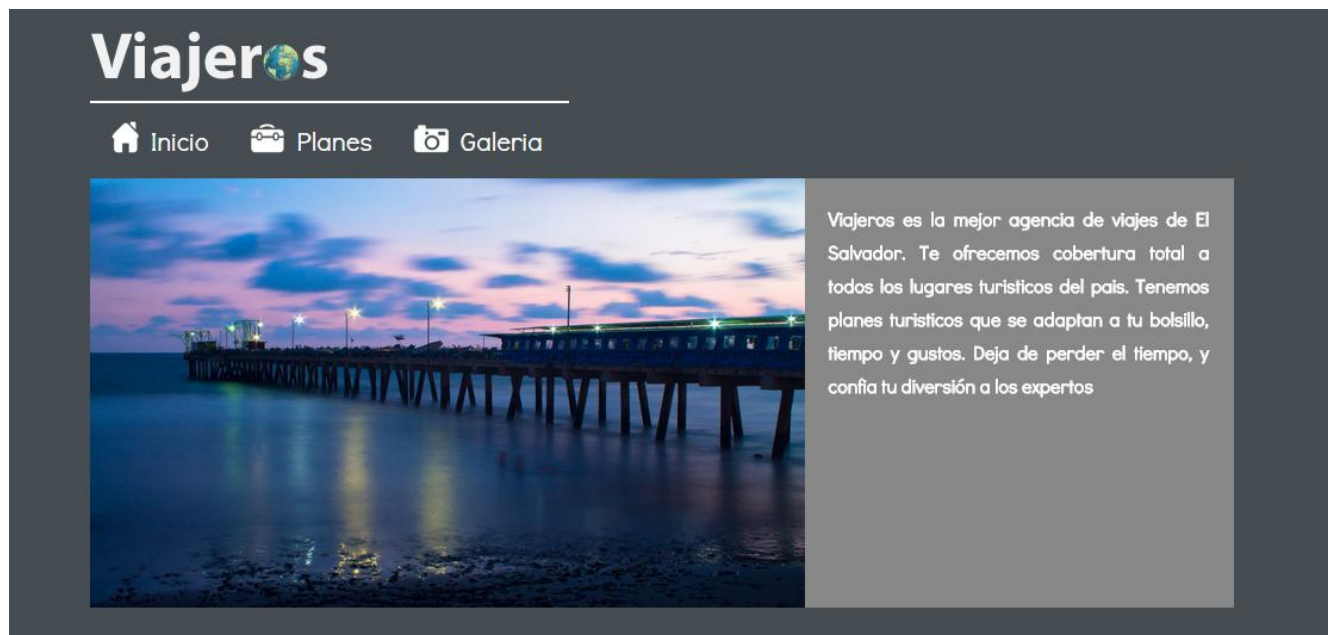


19. Con esto hemos finalizado el menú de navegación. **Cierre el archivo “menú.css” puesto que las siguientes reglas de estilo deberá colocarlas dentro del archivo “fijo.css”.**
20. Ahora finalizaremos el estilo de la sección con id “superior”. Consiga que la imagen que se encuentra dentro del div con 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:



21. 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. 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 elementos flotantes. Resuelva el error y visualice los resultados en su navegador. Debería obtener un resultado similar al siguiente:



22. 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;
}

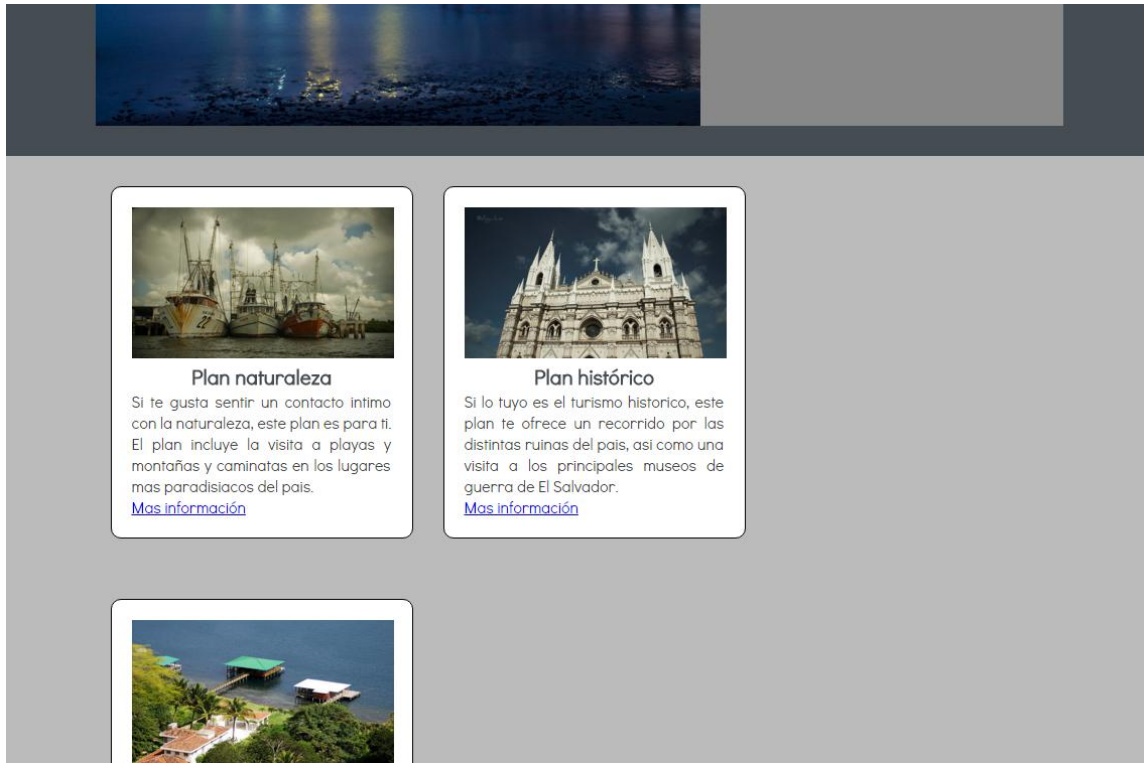
/* PIE DE LA PAGINA*/
footer{
    text-align: center;
    margin: 0;
    font-size: .9em;
    line-height: 2;
    clear: both;
    background-color: #464d52;
    color: white;
    width: 100%;
}
```

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



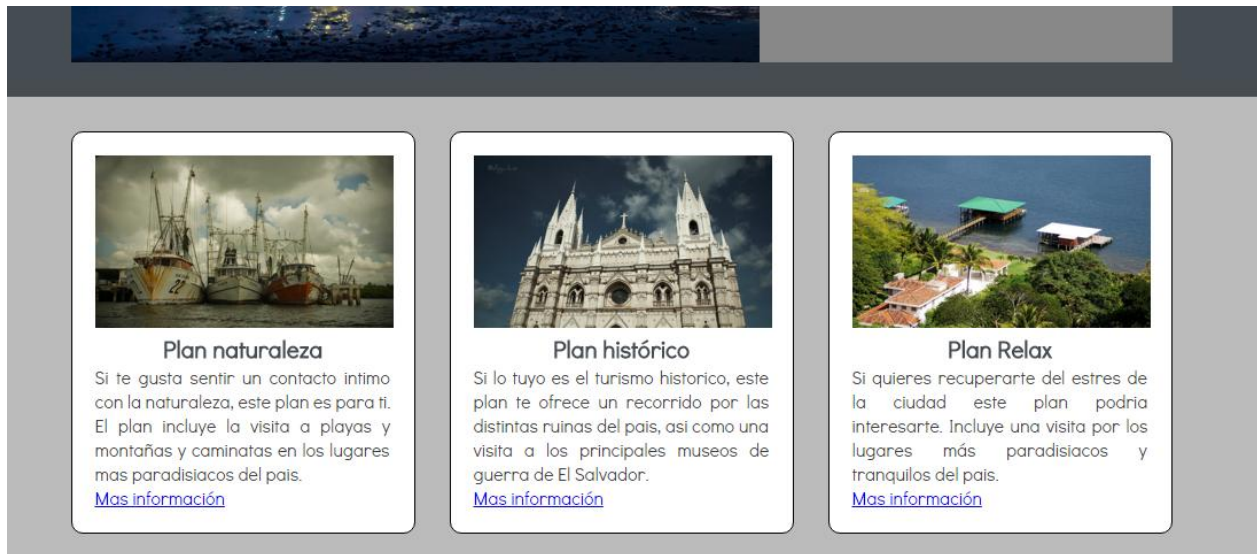
El problema se debe a que el primer div tiene un margen a la izquierda el cual no debería estar. Lo mismo sucede con el tercer div que tiene un margen innecesario a la derecha.

25. 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:



26. 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 página index.html nos servirá como plantilla.

27. Cree una copia de su archivo index.html y guárdela con el nombre planes.html.
28. 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>
```

29. 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

30. Creé una copia de su archivo index.html y guárdela con el nombre galeria.html.
31. Dentro del archivo **galeria.html** elimine todo el contenido de la sección con id "contenido" y reemplázelo 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 Ilopango</p>
  </div>
</div>
```

```

        </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>

```

32. 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:



Iglesia de Ahuachapan



Iglesia de Antigua Cuscatlan



Catedral de San Salvador



Hotel Decameron



Lago de Ilopango



Cementerio de los Ilustres



Puerto de la Libertad



Puerto El Triunfo



Catedral de Santa Ana

IV. Ejercicios complementarios

1. Dentro de los recursos de la guía se le proporciona un archivo llamada “análisis.html”. Utilizar una hoja de estilo externa y aplicar las reglas de estilo necesarias para conseguir los siguientes resultados:

Resultado 1

Usando posicionamiento y z-index



Resultado 2

Usando posicionamiento y z-index

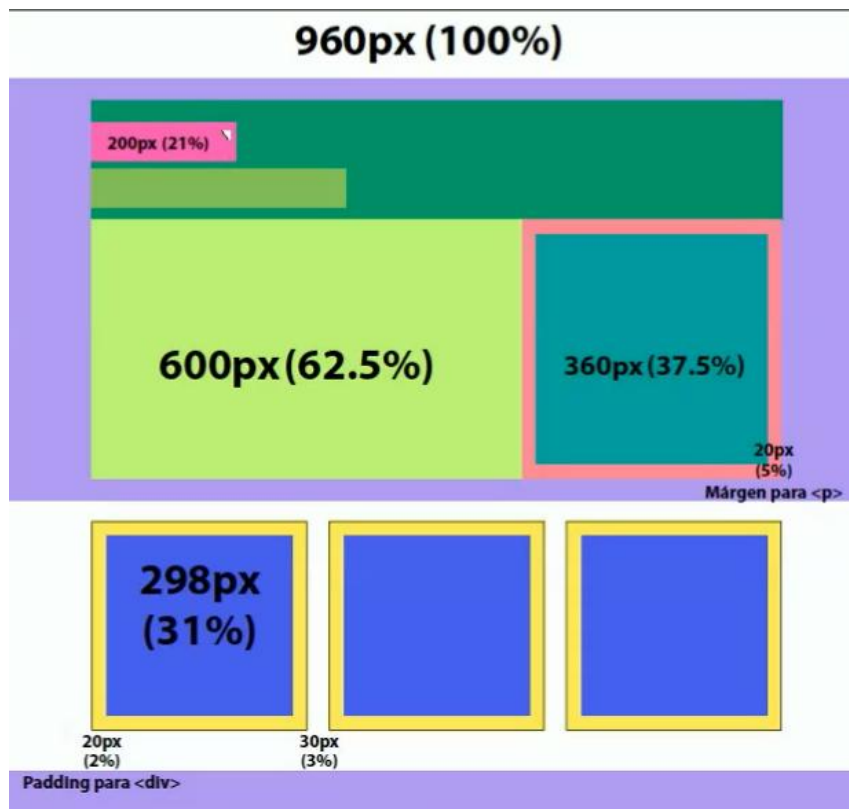


Nota: Debe usar propiedades de posicionamiento y la propiedad z-index

2. 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. Además cambie el float left por un display flex para que las películas se muestren a 3 columnas.

3. Construyendo un diseño líquido

En este ejercicio, vamos a modificar las hojas de estilos ocupadas en el ejercicio 3 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.