	<p style="text-align: center;"><b>UNIVERSIDAD DON BOSCO</b>  <b>FACULTAD DE ESTUDIOS TECNOLOGICOS</b>  <b>ESCUELA DE COMPUTACION</b></p>
<b>CICLO 01</b>	<p style="text-align: center;"><b>GUIA DE LABORATORIO Nº 5</b></p>
	<p><b>Nombre de la practica:</b> Posicionamiento y visualización  <b>Lugar de ejecución:</b> Laboratorio de Informática  <b>Tiempo estimado:</b> 3 horas  <b>Materia:</b> Lenguajes de Marcado y Estilo Web</p>

## 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.

## 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*:

<b>position</b>	Posicionamiento
<b>Valores</b>	<code>static   relative   absolute   fixed   inherit</code>
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	<code>static</code>
<b>Descripción</b>	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:

<b>top</b> <b>right</b> <b>bottom</b> <b>left</b>	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.

<b>display</b>	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.

<b>visibility</b>	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`.

<b>overflow</b>	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:

<b>z-index</b>	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. Requerimientos

Nº	Cantidad	Descripción
1	1	PC con uno de los editores de páginas Web: Sublime Text, Brackets, Notepad ++, BlueFish y HTML Kit y cualquiera de los navegadores siguiente: Internet Explorer 7 u 8, Firefox (última versión disponible), Google Chrome (última versión disponible), Opera (última versión disponible), Safari (última versión disponible).
2	1	Guía de Laboratorio #5 de Lenguajes de Marcado y Estilo Web
3	1	Memoria USB

### IV. Procedimiento

#### Ejercicio 1. Usando posicionamiento y visualización

1. Cuando cargue el sistema operativo, crear una carpeta en Mis Documentos con el nombre: **NºCarnet\_Guia5**, 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. Dar clic en el botón inicio de Windows.
4. Ir a todos los programas y hacer clic en **Sublime Text**.

5. Abrir en Sublime Text la carpeta creada en el paso 1 mediante la opción **"File>Open Folder"** del menú ubicado en la parte superior de la ventana.
6. Crear un nuevo archivo dando click derecho sobre el nombre de la carpeta y seleccionando la opción **"New File"**.
7. Dar clic en la opción **"File>Save"** (o Ctrl+s) para guardar el archivo. Guarde el archivo con el nombre **"oscares.html"**.
8. 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>
```

9. Colocar como título de la página (etiqueta **<title>**) el texto **"Nominaciones mejor película"**.

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

10. Descargar los recursos de la guía de la página web de la universidad y colocar la carpeta "images" en su carpeta de trabajo.
11. 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>

```



12. Después de colocado el texto correspondiente, visualice los resultados de la página en el navegador de su preferencia.
13. Crear un nuevo archivo y guardarlo en la carpeta css con el nombre **style.css**
14. 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">
```

15. 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 importo desde google fonts y se justifica todo el texto de la página.

16. 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”.

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

```

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

```

18. 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%;
}

```

19. 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%;  
}
```

20. Consiga que los elementos article floten a la izquierda, haciendo esto debería conseguir que se muestren la información de las películas en tres columnas.
21. 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;
```

22. 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.
23. 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.

24. 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%;
```

25. 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:



26. 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;
```

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

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

28. ¿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.

29. Consiga que el título de cada una de las películas se muestre centrado usando la propiedad `text-align`.
30. 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

31. 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;  
}
```

32. ¿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 estar seguro de entender este comportamiento antes de seguir con la guía.
33. ¿Qué ha pasado con las otras cajas? ¿Han tomado el espacio que dejó libre el segundo artículo? ¿Por qué?

## Usando posicionamiento absoluto

34. Modifique el posicionamiento del segundo artículo de “relativo” a “absoluto” y visualice los cambios en el navegador.
35. ¿A partir de qué posición se ha desplazo el segundo artículo? El article se ha desplazo 100 pixeles desde la izquierda y 50 pixeles desde arriba a partir del body de la página. Esto ocurre porque el articulo 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.
36. Para cambiar ese comportamiento haga los cambios necesarios para que la sección tenga un posicionamiento relativo y visualice los cambios en el navegador.
37. Notará que ahora el desplazamiento se ha tomado desde el origen de la sección y no desde el origen del body.
38. ¿Qué ha pasado con las otras cajas? ¿Han ocupado el espacio que dejo disponible el segundo artículo?
39. A partir del comportamiento visto, explique ¿Cuáles son las diferencias entre el posicionamiento relativo y absoluto?
40. 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

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

```
visibility: hidden;
```

Visualice los resultados en su navegador.

42. Elimine la declaración anterior y oculte el segundo articulo esta vez usando la siguiente declaración:

```
display: none;
```

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

## V. Análisis de resultados

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

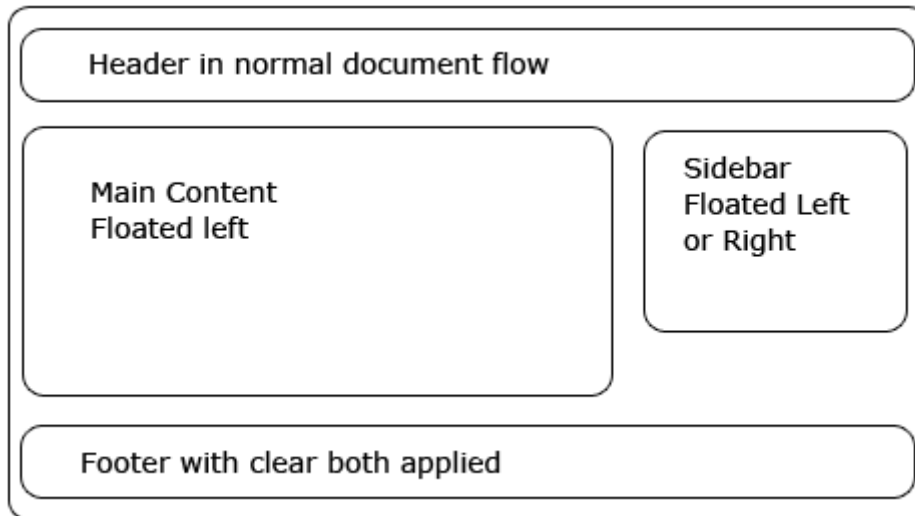


**Nota:** Debe usar propiedades de posicionamiento y la propiedad z-index

El instructor evaluará el desempeño de su trabajo dentro la clase. Estos ejercicios se evaluarán en la misma hora de clase.

## VI. Ejercicios complementarios

1. Modifique el ejemplo 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. 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.