

3.4.6. Posicionamiento

CSS tiene 5 formas de establecer la posición en pantalla de una caja:

Estático, Relativo, Absoluto, Fijo y Flotante

Para establecer la posición disponemos de las propiedades:

position, top, bottom, right, left y float

Propiedad: **position**

Significado: **establece la posición de la caja en pantalla**

Valores: **static | relative | absolute | fixed | inherit**

Aplicación: **a todos los elementos**

El valor por defecto es static

- static: la caja ignora las propiedades top, bottom, right y left
- relative: la caja se desplazará según los valores top, bottom, right y left
- absolute: la caja se desplazará según los valores top, bottom, right y left partiendo de su elemento contenedor
- fixed: como el absoluto pero la caja permanece inmóvil independientemente del resto de elementos y del movimiento de la página por parte del usuario
- inherit: heredado

Propiedades: **top, bottom, right, left**

Significado: **establece el desplazamiento vertical y horizontal de la caja.**

Valores: **<número px> | <número %> | auto | inherit**

Aplicación: **a todos los elementos**

El valor por defecto es auto

Propiedad: **float**

Significado: **establece la posición flotante de la caja**

Valores: **left | right | none | inherit**

Aplicación: **a todos los elementos**

El valor por defecto es none

· **Posicionamiento estático.** {position: static}

Se trata del posicionamiento por defecto que utilizan todos los navegadores.

Si los elementos a posicionar son de bloque, como un párrafo, las cajas se colocarán verticalmente una debajo de otra, dentro del elemento contenedor.

Si los elementos a posicionar son de línea, como un hipervínculo, las cajas se colocarán una junto a otra horizontalmente dentro del elemento de bloque.

Si las cajas no caben en una línea, entonces ocuparán la siguiente, etc. y si caben en una línea, se pueden alinear horizontalmente.

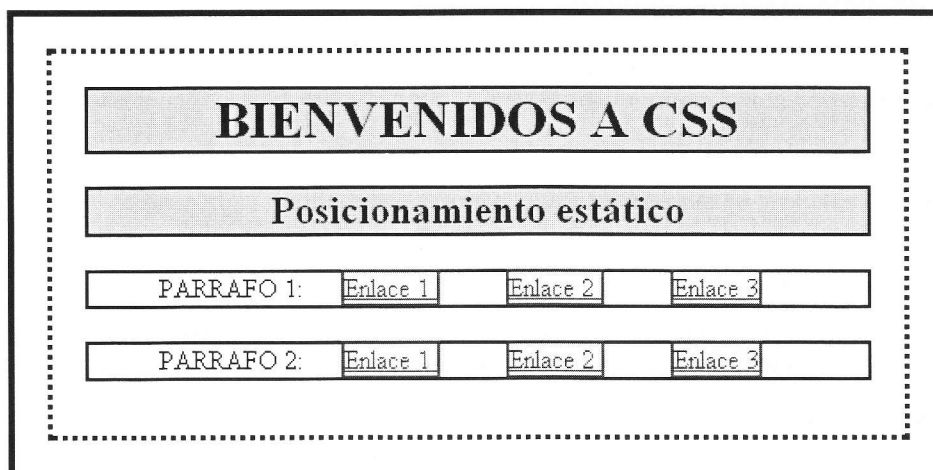


Figura 3.11: Posicionamiento estático

```
<style>
  * {border: 2px solid; text-align: center; margin: 20px;}
  html {border: 4px solid;}
  body {border: dotted;}
  h1,h2 {background-color: pink;}
  a {background-color: yellow;}
</style>

<body>
  <h1>BIENVENIDOS A CSS</h1>
  <h2>Posicionamiento Estático</h2>
  <p> PARRAFO 1:
    <a href="#"> Enlace 1 </a> <a href="#"> Enlace 2 </a>
    <a href="#"> Enlace 3 </a>
  </p>
  <p> PARRAFO 2:
    <a href="#"> Enlace 1 </a><a href="#"> Enlace 2 </a>
    <a href="#"> Enlace 3 </a>
  </p>
</body>
```

· **Posicionamiento relativo.** {position: relative}

Consiste en desplazar la caja una cantidad respecto del posicionamiento estático.

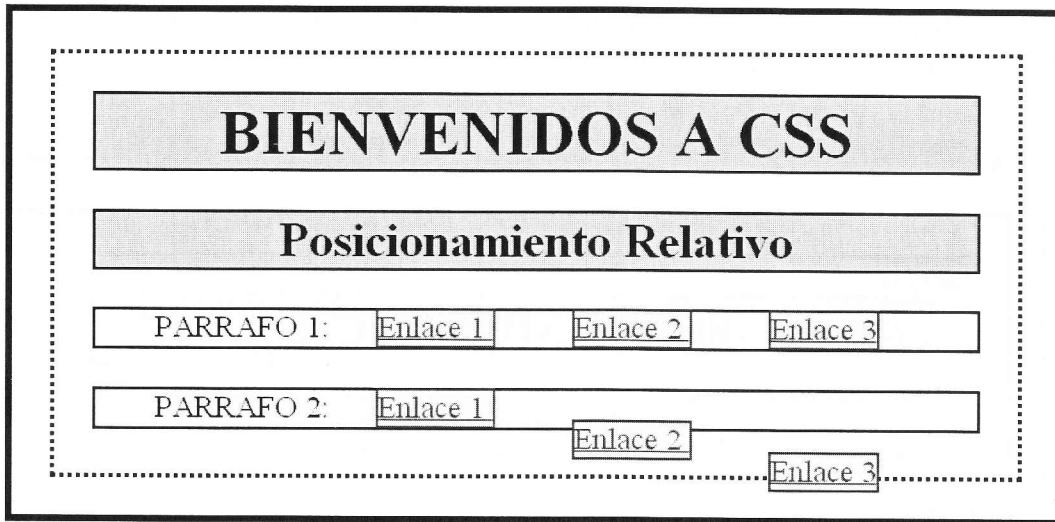


Figura 3.12: Posicionamiento relativo

```
<style>
  * {border: 2px solid; text-align: center; margin: 20px;}
  html {border: 4px solid;}
  body {border: dotted;}
  h1,h2 {background-color: pink;}
  a {background-color: yellow;}
  a.abajo {position: relative; top: 1em;}
  a.mas_abajo {position: relative; top: 2em;}
</style>

<body>
  <h1>BIENVENIDOS A CSS</h1>
  <h2>Posicionamiento Relativo</h2>
  <p>
    PARRAFO 1:
    <a href="#"> Enlace 1 </a>
    <a href="#"> Enlace 2 </a>
    <a href="#"> Enlace 3 </a>
  </p>
  <p>
    PARRAFO 2:
    <a href="#"> Enlace 1 </a>
    <a href="#" class="abajo"> Enlace 2 </a>
    <a href="#" class="mas_abajo"> Enlace 3 </a>
  </p>
</body>
```

Como se puede observar, la caja “Enlace 2” se ha desplazado 1em hacia abajo desde su posición normal (static), y la caja “Enlace 3” se ha desplazado 2em.

· **Posicionamiento absoluto.** {position: absolute}

Consiste en desplazar la caja una cantidad respecto de su elemento contenedor.

La caja sale del flujo normal de la página y el resto de cajas ocupan su lugar.

Para el desplazamiento se usan top, bottom, left right y se toma como origen de coordenadas la esquina superior izquierda del primer contenedor que no sea static.

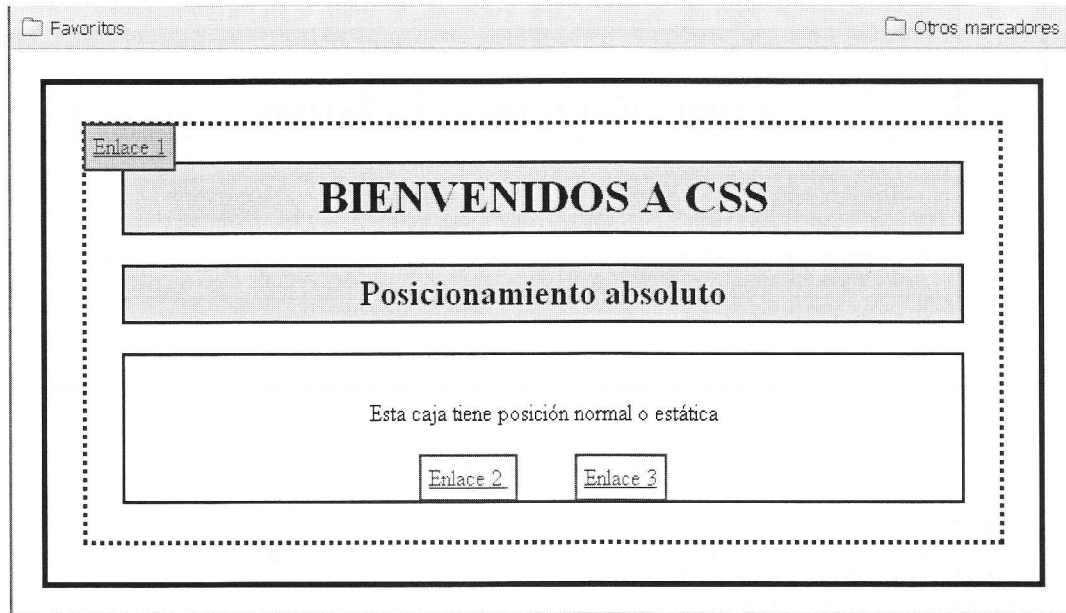


Figura 3.13: Posicionamiento absoluto

```
<style>
  * {text-align: center; margin: 20px; padding: 5px;}
  html {border: 4px solid;}
  body {border: dotted;}
  h1,h2 {background-color: pink; border: 2px solid;}
  div {border: 2px solid;}
  a {background-color: yellow; border: 2px solid;}
  a.absoluto {background-color: orange;
              position: absolute; top: 30px; left: 30px;}
</style>

<body>
  <h1>BIENVENIDOS A CSS</h1>
  <h2>Posicionamiento absoluto</h2>
  <div>
    <p>Esta caja tiene posición normal o estática</p>
    <a href="#" class="absoluto"> Enlace 1 </a>
    <a href="#"> Enlace 2 </a>
    <a href="#"> Enlace 3 </a>
  </div>
</body>
```

En este caso el “Enlace1” se desplaza 30px desde arriba y desde la izquierda, partiendo desde la esquina superior izquierda de la página, porque `<div>` tiene posición normal y el siguiente es `<body>` (observar que además hay 20px de margen).

Las cajas “Enlace2” y “Enlace3” ocupan el hueco dejado por la caja “Enlace1”.

Veamos otro ejemplo de posicionamiento absoluto.

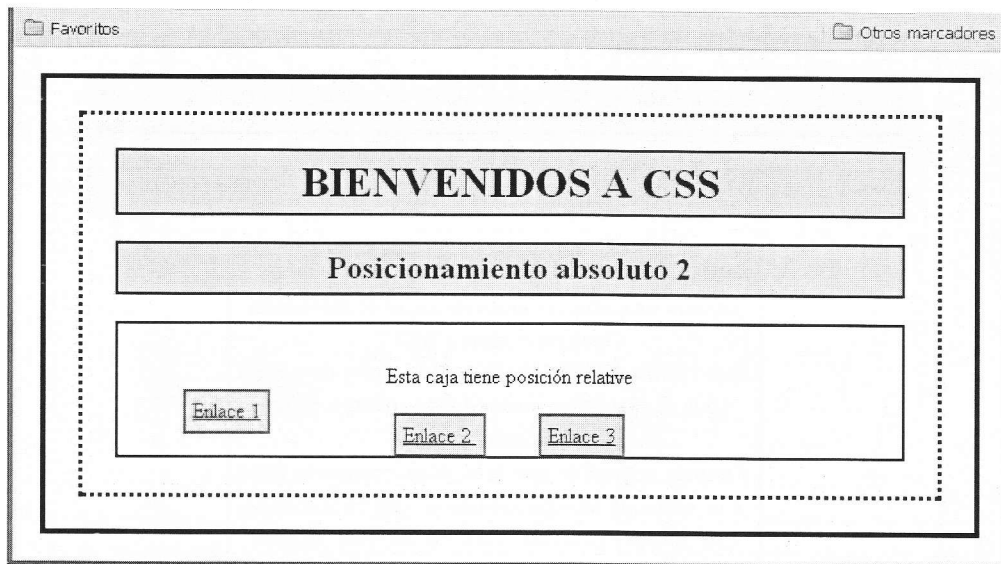


Figura 3.14: Posicionamiento absoluto

```
<style>
  * {text-align: center; margin: 20px; padding: 5px;}
  html {border: 4px solid;}
  body {border: dotted;}
  h1,h2 {background-color: pink; border: 2px solid;}
  div {position: relative; border: 2px solid;}
  a {background-color: yellow; border: 2px solid;}
  a.absoluto {background-color: orange;
              position: absolute; top: 30px; left: 30px;}
</style>
```

En este caso el “Enlace1” se desplaza 30px desde arriba y desde la izquierda, pero partiendo de su caja contenedora, que es la caja `<div>`, porque es la primera caja con posicionamiento no estático (tiene posición relative).

· **Posicionamiento fijo.** {position: fixed}

Similar al posicionamiento absoluto, pero cuando el usuario desplaza la página las cajas se mantienen fijas en el mismo lugar.

Esto resulta útil para encabezados o pies de página que se mantienen inalterables a lo largo de un documento extenso.

También es posible aplicar el posicionamiento fixed a un menú de navegación lateral, que necesitamos esté siempre presente, para los famosos widgets sociales, o un pequeño menú de ayuda al usuario.

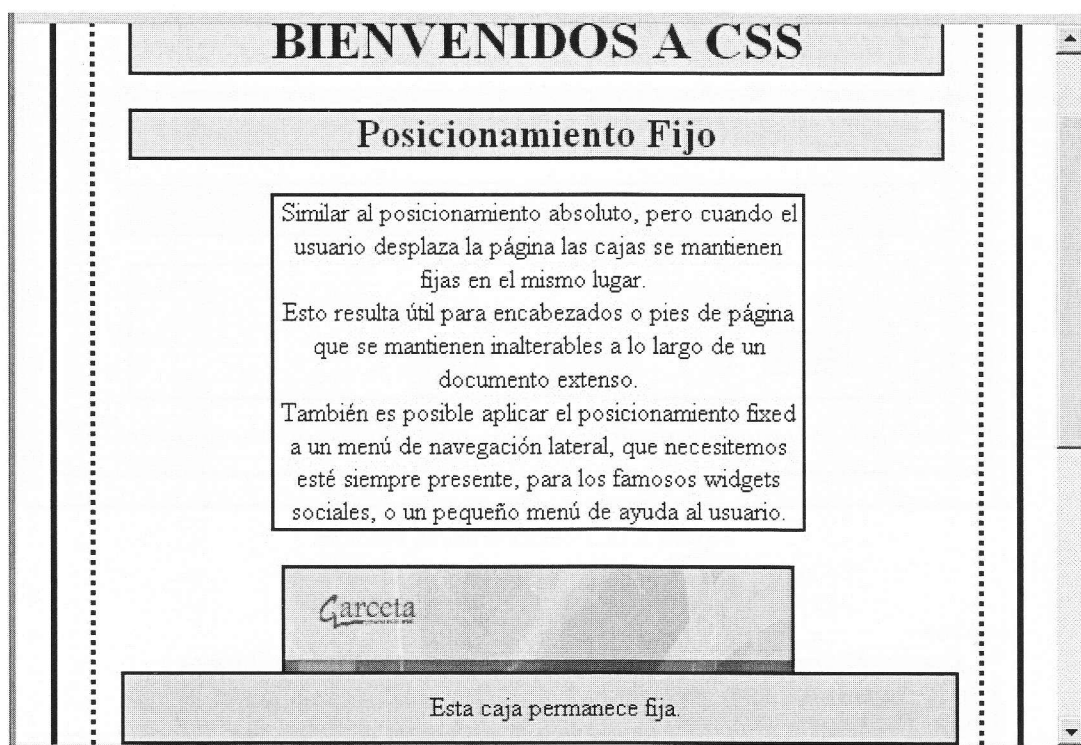


Figura 3.15: Posicionamiento fijo

```
<style>
  * {border: 2px solid; text-align: center; margin: 20px;}
  html {border: 4px solid;}
  body {border: dotted;}
  h1,h2 {background-color: pink;}
  p {width: 60%; position: relative; left: 20%; margin: 0;}
  .fijo {position: fixed; bottom:0; width: 80%; height: 40px;
        margin: 0; left: 10%; background-color:#CCC}
  .fijo p {border: 0; margin: 10px;}
</style>
```

Actividad 3.4:

Escribe el código html que falta para obtener la figura 3.10.

Comprueba como la caja inferior permanece fija aunque utilices la barra de desplazamiento.

· **Posicionamiento flotante.** {float: left | right | inherit | none}

- Desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.
- El resto de cajas ocupan el lugar dejado por la caja flotante.
- No se produce solapamiento de cajas porque la propiedad float tiene en cuenta el resto de cajas flotantes.

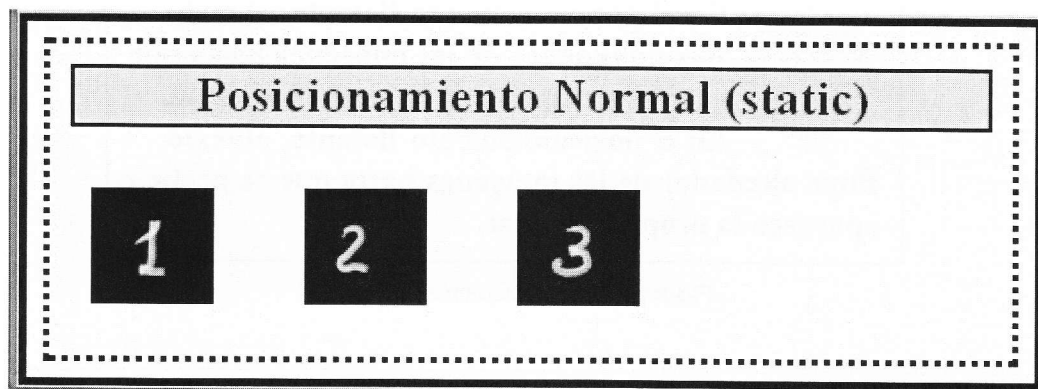


Figura 3.16: Posicionamiento normal

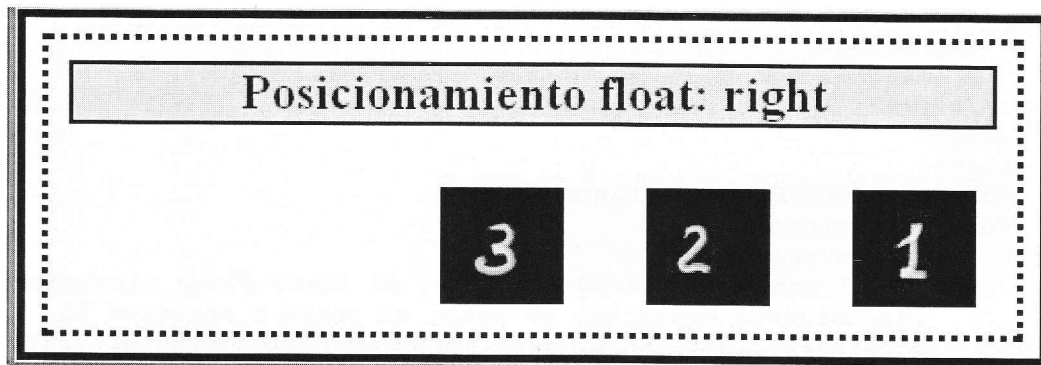


Figura 3.17: Posicionamiento float: right

Con **float: right** las imágenes se desplazan todo lo posible hacia la derecha de su posición original pero respetando al resto de cajas.

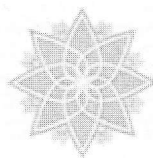
```

<style>
  html {border: 4px solid;}
  body {border: 2px dotted;}
  h2 {background-color: pink; border: 2px solid;
    margin: 10px;text-align: center;}
  img {width: 60px; margin: 20px; float: right;}
</style>

```

- Los elementos situados alrededor de la caja flotante se adaptan a la posición de la caja, lo que hace que sea idónea para la combinación de texto con imágenes.

Posicionamiento float: left



En el posicionamiento flotante, el texto fluye alrededor de las imágenes hasta que se acabe o aparezca la propiedad clear.

En el posicionamiento flotante, el texto fluye alrededor de las imágenes hasta que se acabe o aparezca la propiedad clear.

Figura 3.18: Posicionamiento float: left

```

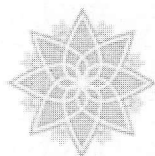
<html>
<head>
  <title> Posicionamiento flotante</title>
  <style>
    #principal {border: 2px solid; width: 400px; font-size: large;}
    img {float: left;}
    p {margin: 5px;}
  </style>
</head>
<body>
  <h1>Posicionamiento flotante</h1>
  <div id="principal">
    
    <p> En el posicionamiento flotante, el texto fluye alrededor de
      las imágenes hasta que se acabe el texto o aparezca la
      propiedad clear. En el posicionamiento flotante, el texto
      fluye alrededor de las imágenes hasta que se acabe el texto o
      aparezca la propiedad clear.
    </p>
  </div>
</body>
</html>

```


- La propiedad `clear` permite evitar el comportamiento del posicionamiento flotante. En el valor se establece el lado que no debe ser adyacente a ninguna caja flotante.

```
{ clear: none | left | right | both | inherit }
```

Propiedad clear



La propiedad `clear` evita el posicionamiento flotante y por tanto, el texto se coloca inmediatamente debajo de la imagen, en lugar de rodearla.

Figura 3.19: Propiedad clear

3.4.7. Visualización

En CSS hay cuatro propiedades que controlan la visualización de las cajas: `display`, `visibility`, `z-index` y `overflow`.

Propiedad: **display**

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
```

Significado:

- El valor **inline** hace que cualquier elemento html se muestre como elemento en línea.

Se suele utilizar en las listas que se quieren mostrar horizontalmente.

- El valor **block** hace que cualquier elemento html se muestre como elemento de bloque, es decir ocupa todo el ancho de línea disponible.

Se suele utilizar para los enlaces que forman el menú de navegación.

Vamos a combinar las propiedades `background-color` y `display: block`, con el selector `a:hover` para crear un menú de navegación.

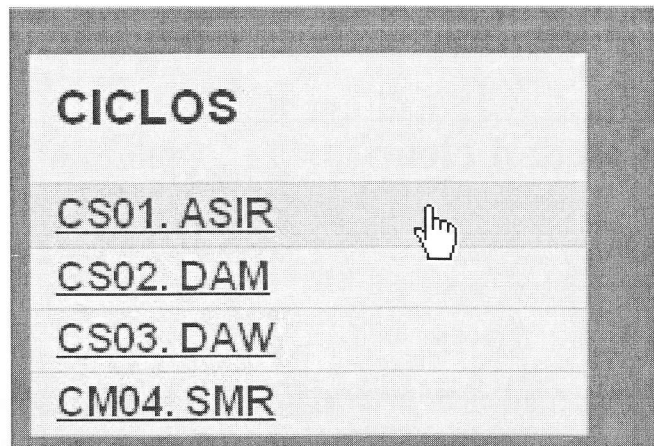


Figura 3.20: Menú de navegación

```
<style>
  body {background-color: #996;}
  a:link {color: #006699; padding: 2px 0px 2px 10px;
    border-top: 1px solid #cccccc; display: block;}
  a:hover {color: #006699; border-top: 1px solid #cccccc;
    background-color: #dddddd; padding: 2px 0px 2px 10px;}
  h3 {color: #334d55; padding: 10px 0px 0px 10px;}
  #navBar {width: 220px; height: auto; margin-top: 10px;
    padding: 0px; background-color: #eeeeee; font-family: arial;
    border-bottom: 1px solid #ccc; border-right: 1px solid #ccc;
    font-weight: bold;}
</style>
...
<body>
  <div id="navBar">
    <h3> MODULOS </h3>
    <a href="asir.html">CS01. ASIR</a>
    <a href="dam.html">CS02. DAM</a>
    <a href="daw.html">CS03. DAW</a>
    <a href="smr.html">CM04. SMR</a>
  </div>
</body>
```

La propiedad `display: block` nos permite hacer que un elemento en línea como `<a>` se comporte como un elemento de bloque y por tanto, ocupe toda la línea en la que se encuentra.

Mediante `a:hover` y `background-color: #ddd` conseguimos que la línea cambie de color cuando situamos el ratón encima, dando la seguridad al usuario de que ha seleccionado la opción deseada.