

# 42 CSS: Ventana Modal / Menú Móvil con `position: fixed`

El posicionamiento fijo (`position: fixed`) es extremadamente útil para crear elementos de interfaz de usuario que necesitan **superponerse al contenido principal** de la página y **permanecer visibles o en una posición constante** independientemente del scroll. Dos ejemplos clásicos de esto son las ventanas modales y los menús de navegación móviles (a menudo de tipo "off-canvas" o deslizables).

## 1. Ventana Modal (Popup / Dialog)

Una ventana modal es un cuadro de diálogo o panel que aparece encima del contenido principal de la página, requiriendo la interacción del usuario antes de poder volver al contenido subyacente. A menudo se usa para formularios importantes, alertas, galerías de imágenes, etc.

- **Componentes Típicos:**

- **Overlay/Fondo:** Un fondo semitransparente que cubre toda la ventana gráfica, indicando que el contenido detrás está inactivo.
- **Modal Box/Contenedor:** La caja que contiene el contenido real de la modal (texto, formulario, botones).

- **¿Por qué `position: fixed` ?**

- **Overlay:** Debe cubrir toda la pantalla sin importar el scroll. `position: fixed` con `top: 0; left: 0; right: 0; bottom: 0;` es perfecto para esto.
- **Modal Box:** Debe aparecer encima del overlay y también permanecer fija en la pantalla, usualmente centrada. `position: fixed` permite esto.

- **Ejemplo de Estructura (HTML):**

```
<!-- Botón para abrir la modal (controlado por JS) →
<button id="openModalBtn">Abrir Modal</button>

<!-- La modal (inicialmente oculta) →
<div class="modal-overlay" id="myModal">
  <div class="modal-content">
    <span class="close-btn">&times;</span> <!-- Botón para cerrar →
    <h2>Título de la Modal</h2>
    <p>Este es el contenido de la ventana modal.</p>
    <button>Aceptar</button>
  </div>
</div>
```

- **Ejemplo de Estilos (CSS):**

```
/* Overlay */
.modal-overlay {
  display: none; /* Inicialmente oculto - JS lo mostrará */
```

```

position: fixed; /* Fijo al viewport */
top: 0;
left: 0;
right: 0;
bottom: 0;
background-color: rgba(0, 0, 0, 0.6); /* Negro semitransparente */
z-index: 1050; /* Alto z-index para estar sobre el contenido */
/* Opcional: usar flex/grid para centrar el .modal-content */
display: flex;
align-items: center;
justify-content: center;
}

/* Contenido de la Modal */
.modal-content {
  /* Si no se usa flex/grid en el overlay para centrar: */
  /* position: fixed; */
  /* top: 50%; */
  /* left: 50%; */
  /* transform: translate(-50%, -50%); */ /* Técnica de centrado absoluto */

  background-color: white;
  padding: 20px 30px;
  border-radius: 5px;
  min-width: 300px; /* Ancho mínimo */
  max-width: 80%; /* Ancho máximo */
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.2);
  z-index: 1051; /* Encima del overlay */
  position: relative; /* Para posicionar el botón de cerrar absoluto */
}

/* Botón de cerrar (ejemplo) */
.close-btn {
  position: absolute;
  top: 10px;
  right: 15px;
  font-size: 24px;
  font-weight: bold;
  color: #aaa;
  cursor: pointer;
}
.close-btn:hover {
  color: #333;
}

/* Clase para mostrar la modal (añadida/quitada con JS) */
/* .modal-overlay.active {

```

```
display: flex; O block, si no usas flex/grid para centrar
} */
```

- **Interacción:** CSS define cómo se ve y se posiciona la modal y su overlay. **JavaScript** es necesario para controlar la visibilidad (mostrar/ocultar la modal al hacer clic en botones, usualmente añadiendo/quitando una clase `active` que cambia `display` de `none` a `flex` o `block`).

## 2. Menú Móvil (Off-Canvas / Deslizable)

En pantallas pequeñas, es común ocultar la navegación principal y mostrarla solo cuando el usuario toca un botón (ícono de hamburguesa). El menú a menudo se desliza desde un lado de la pantalla.

- **Componentes Típicos:**
  - **Contenedor del Menú:** El elemento que contiene los enlaces de navegación.
  - **Botón de Toggle:** El ícono (hamburguesa) para abrir/cerrar.
  - **(Opcional) Overlay:** A veces se usa un overlay similar al de la modal para oscurecer el contenido principal cuando el menú está abierto.
- **¿Por qué `position: fixed` ?**
  - El menú debe aparecer **encima** del contenido de la página.
  - Debe permanecer **fijo** en su posición (generalmente pegado a un lado de la pantalla) incluso si el contenido detrás se puede desplazar.
- **Ejemplo de Estructura (HTML):**

```
<button class="menu-toggle">≡</button>

<nav class="mobile-nav" id="mainNav">
  <ul>
    <li><a href="#">Inicio</a></li>
    <li><a href="#">Servicios</a></li>
    <li><a href="#">Contacto</a></li>
  </ul>
</nav>
```

- **Ejemplo de Estilos (CSS):**

```
.mobile-nav {
  position: fixed; /* Fijo al viewport */
  top: 0;
  left: 0; /* O right: 0; si quieres que salga de la derecha */
  width: 250px; /* Ancho del menú */
  height: 100%; /* Altura completa del viewport */
  background-color: #333;
  padding-top: 60px; /* Espacio para contenido o logo */
  box-sizing: border-box;
  z-index: 1000;

  /* Estado inicial: Oculto fuera de la pantalla a la izquierda */
}
```

```

transform: translateX(-100%);
/* Transición suave para el deslizamiento */
transition: transform 0.3s ease-in-out;
}

/* Estado activo: Visible en pantalla */
.mobile-nav.is-open {
transform: translateX(0);
}

/* Estilos básicos para los enlaces (ejemplo) */
.mobile-nav ul {
list-style: none;
padding: 0;
margin: 0;
}
.mobile-nav li a {
display: block;
padding: 15px 20px;
color: white;
text-decoration: none;
border-bottom: 1px solid #444;
}
.mobile-nav li a:hover {
background-color: #555;
}

/* El botón de toggle necesita sus propios estilos */
.menu-toggle {
position: fixed; /* O absolute respecto a un header */
top: 15px;
left: 15px;
z-index: 1001; /* Encima del menú si es necesario */
/* ... otros estilos para el botón */
}

```

- **Interacción:** Similar a la modal, CSS define la apariencia, posición y la animación de transición ( `transform` y `transition` ). **JavaScript** se encarga de detectar el clic en el botón `.menu-toggle` y añadir/quitar la clase `is-open` al `.mobile-nav` para activar la transición y mostrar/ocultar el menú.

#### Consideraciones Comunes:

- **z-index** : Es crucial para asegurar que la modal/menú y su overlay (si existe) aparezcan por encima del resto del contenido de la página.
- **Bloqueo de Scroll:** A menudo, cuando una modal o un menú off-canvas está abierto, se desea evitar que el contenido del `body` detrás se pueda desplazar. Esto generalmente se logra con JavaScript, añadiendo temporalmente `overflow: hidden;` al `<body>` o al elemento contenedor principal.

- **Accesibilidad:** Es vital considerar la accesibilidad:
  - Manejo del foco: El foco del teclado debe moverse a la modal/menú cuando se abre y volver al elemento que lo activó cuando se cierra.
  - Cierre: Debe ser posible cerrar con la tecla Escape y/o un botón de cierre claro.
  - ARIA: Usar atributos ARIA apropiados ( `role="dialog"` , `aria-modal="true"` , etc.) para informar a las tecnologías de asistencia.

### Conclusión:

`position: fixed` es la base para crear elementos de interfaz superpuestos y persistentes como ventanas modales y menús móviles. Permite posicionar estos elementos respecto a la ventana gráfica, asegurando que permanezcan visibles y en su lugar independientemente del scroll. Combinado con `z-index` para el apilamiento y JavaScript para controlar la visibilidad y la interacción, `position: fixed` nos da las herramientas necesarias para construir estos patrones de UI tan comunes y útiles.

---