



TAREA 4: ANÁLISIS DE SITIOS WEB CON CONTENIDO MULTIMEDIA

Revisar los siguientes sitios web y comentar cuál os parece mejor y por qué.

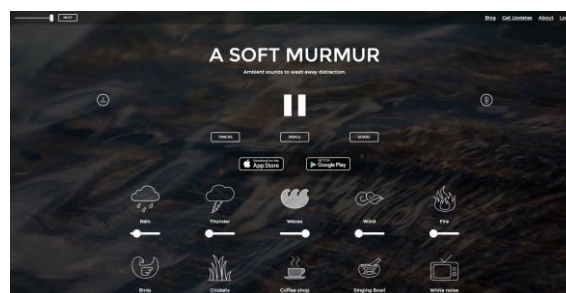
- <http://www.donothingfor2minutes.com/>
- <https://asoftmurmur.com/>
- <https://www.meditation.com/>
- <https://mynoise.net/>

SITIO WEB 1: <http://www.donothingfor2minutes.com/>



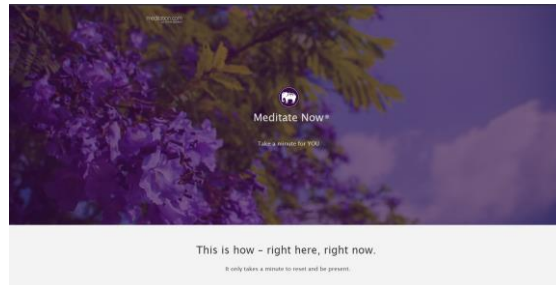
De las cuatro páginas web aportadas es el que menos me ha gustado. Tiene un diseño muy simple y, aunque entiendo que el objetivo de la web es que no hagas nada durante 2 minutos y te relajes escuchando el sonido de las olas, en comparación con el resto de las páginas me parece que no termina de conseguir el efecto deseado (por ejemplo, el simple hecho de ajustar el volumen hace que falles y puede resultar algo frustrante).

SITIO WEB 2: <https://asoftmurmur.com/>



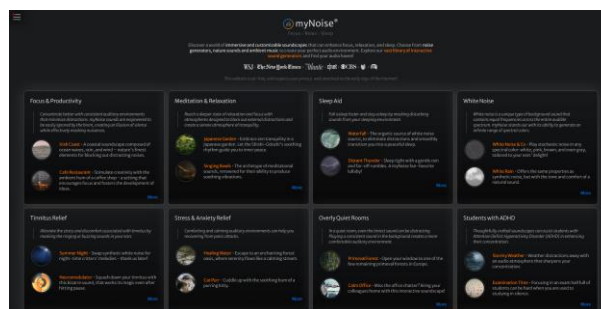
Este sitio es el segundo que más me ha gustado porque tiene varias opciones para personalizar el sonido que más se acomode a tus gustos personales (pájaros, cafetería, crepitar del fuego...), además a diferencia del anterior permite el que inicies el sonido y puedas realizar otras tareas en el ordenador mientras tienes el sonido de fondo.

SITIO WEB 3: <https://www.meditation.com/>



Me parece una buena página para meditar y relajarse, aunque el que siempre sea el mismo sonido puede resultar algo monótono, faltaría que hubiera cierta personalización más allá de la duración de la sesión de meditación. En cuanto a diseño me parece bastante simple pero efectivo en su objetivo.

SITIO WEB 4: <https://mynoise.net/>

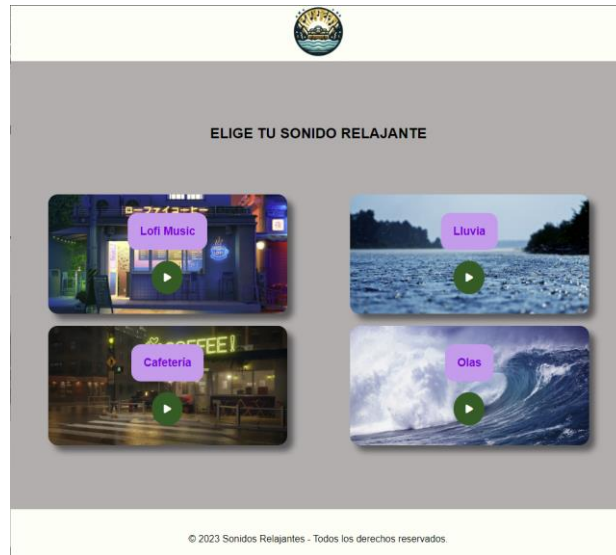


Conocía esta página de antes y es de hecho una de las páginas que suelo utilizar para tener algo de sonido de fondo para relajarme mientras leo, estudio o realizo programación ya que me ayuda a aumentar la concentración. La cantidad de opciones de personalización es mucho mayor que en las páginas anteriores y no sólo cuentas con ambientes predefinidos como una cascada, una tormenta, un jardín zen... sino que una vez dentro de cada uno puedes utilizar un ecualizador para personalizar cada ambiente según tus gustos.

A nivel de diseño y usabilidad me parece la mejor web de las cuatro aportadas en la tarea.

Os propongo realizar una web similar a las anteriores (este punto es voluntario) en su mínima expresión (sonido ambiental configurable o con algún video en plan relax).

He realizado la siguiente web:



Se compone de 4 elementos section que almacenan en su interior un elemento article con un h3 y un icono. Este es el main del documento HTML:

```
<main class="contenedor-flex">
  <h2>Elige tu sonido relajante</h2>
  <section class="contenedor-grid">
    <section id="lofi">
      <article>
        <h3>Lofi Music</h3>
        <i id="btn-lofi" class="fa-solid fa-play"></i>
      </article>
    </section>
    <section id="lluvia">
      <article>
        <h3>Lluvia</h3>
        <i id="btn-lluvia" class="fa-solid fa-play"></i>
      </article>
    </section>
    <section id="cafeteria">
      <article>
        <h3>Cafeteria</h3>
        <i id="btn-cafeteria" class="fa-solid fa-play"></i>
      </article>
    </section>
    <section id="olas">
      <article>
        <h3>Olas</h3>
        <i id="btn-olas" class="fa-solid fa-play"></i>
      </article>
    </section>
  </section>
</main>
```

A este documento le acompaña una hoja de estilado para darle formato a la página, por ejemplo, se utiliza una rejilla grid para colocar los elementos section en una cuadrícula de 2x2 y se añade un cambio de formato en dispositivos de máximo 900 px de ancho:

```
* {
  padding: 0;
  margin: 0;
  box-sizing: border-box;
}

body {
  font-family: "Arial", sans-serif;
}

h2 {
  text-transform: uppercase;
}

h3 {
  color: blueviolet;
  background: #ceaeef;
  padding: 20px;
  border-radius: 16px;
}

header, footer {
  display: flex;
  justify-content: center;
  align-items: center;
  background-color: #fdfdff;
  min-height: 100px;
}

header img {
  width: 100px;
}

.contenedor-flex {
  width: 100%;
  display: flex;
  justify-content: space-evenly;
  align-items: center;
  flex-flow: column nowrap;
  min-height: 750px;
  padding: 20px;
  background-color: #b2aeae;
}

.contenedor-grid {
  display: grid;
  grid-template-rows: repeat(2, 1fr);
  grid-template-columns: repeat(2, 1fr);
  width: 100%;
  gap: 20px;
  justify-items: center;
  align-items: center;
}

section>section {
  display: flex;
  justify-content: space-evenly;
  align-items: center;
  flex-flow: column nowrap;
  width: 400px;
  height: 200px;
  border-radius: 16px;
  box-shadow: 10px 10px 10px rgba(0, 0, 0, 0.5);
}

article {
  display: flex;
  justify-content: space-evenly;
  align-items: center;
  flex-flow: column nowrap;
  height: 170px;
}

.contenedor-grid section:first-child {
  background: url(.../images/lofi.jpg);
  background-size: 100%;
}

.contenedor-grid section:nth-child(2) {
  background: url(.../images/lluvia.jpg);
  background-size: 100%;
}

.contenedor-grid section:nth-child(3) {
  background: url(.../images/cafeteria.png);
  background-size: 100%;
}

.contenedor-grid section:nth-child(4) {
  background: url(.../images/olas.jpg);
  background-size: 100%;
}
```

```
i {
  display: block;
  background-color: #195A22FF;
  color: white;
  font-size: 1rem;
  padding: 20px;
  border-radius: 50%;
  margin: 1%;
  transition: 0.5s;
}

i:hover {
  cursor: pointer;
}

i:active {
  transform: scale(0.5);
}

.sonidoActivado {
  background-color: #FC3232FF;
}

@media screen and (max-width: 900px) {
  .contenedor-grid {
    display: flex;
    width: 100%;
    gap: 20px;
    justify-items: center;
    align-items: center;
    flex-flow: column nowrap;
  }
}
```

Por último, se añade un documento JS para darle funcionalidad a la página con las siguientes funciones:

```
/**
 * Crea un elemento de audio con el nombre proporcionado y lo agrega al
 * cuerpo del documento HTML.
 * @param {string} nombreBoton - El nombre del botón para generar la ruta
 * del archivo de audio.
 */
1+ usages new *
function addElementoMusica(nombreBoton) {
    const audio = document.createElement('audio');
    audio.id = 'audio';
    audio.loop = true;
    audio.autoplay = true;
    audio.src = `./audio/${nombreBoton}.mp3`;

    document.body.appendChild(audio);
}

/**
 * Elimina el elemento de audio del cuerpo del documento HTML, si existe.
 */
1+ usages new *
function removeElementoAudio() {
    const audio = document.getElementById('audio');

    if (audio) {
        document.body.removeChild(audio);
    }
}

/**
 * Cambia la apariencia de un botón para indicar que la música está activada.
 * @param {string} idBoton - El ID del botón que se modificará.
 */
1+ usages new *
function cambiaBoton(idBoton) {
    const boton = document.getElementById(idBoton);
    boton.setAttribute('class', '');
    boton.classList.add('fa-solid');
    boton.classList.add('fa-stop');
    boton.classList.add('sonidoActivado');
}

/**
 * Formatea todos los botones para indicar que la música está desactivada.
 */
1+ usages new *
function formateaBotones() {
    const botones = document.getElementsByTagName('i');

    for (const boton of botones) {
        boton.setAttribute('class', '');
        boton.classList.add('fa-solid');
        boton.classList.add('fa-play');
    }
}
```

```
/**
 * Controla el evento de clic en los botones para activar/desactivar la música.
 * @param {Event} evento - El evento de clic en un botón.
 */
1+ usages new *
function toggleMusic(evento) {
    const elementoEvento = evento.target.localName;

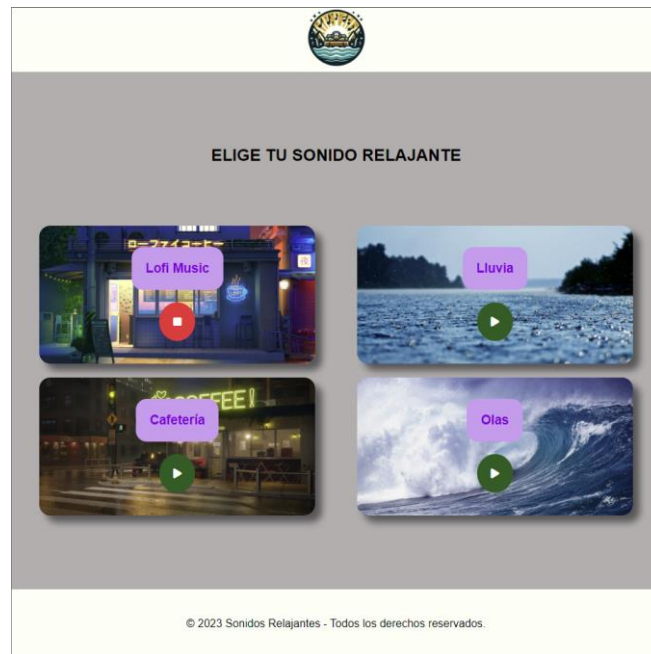
    if (elementoEvento === 'i') {
        const idBoton = evento.target.id;
        const boton = document.getElementById(idBoton);

        if (boton.classList.contains('fa-stop')) {
            formateaBotones();
            removeElementoAudio();
        } else {
            const nombreBoton = evento.target.id.split('-')[1];
            formateaBotones();
            cambiaBoton(evento.target.id);
            removeElementoAudio();
            addElementoMusica(nombreBoton);
        }
    }
}
```

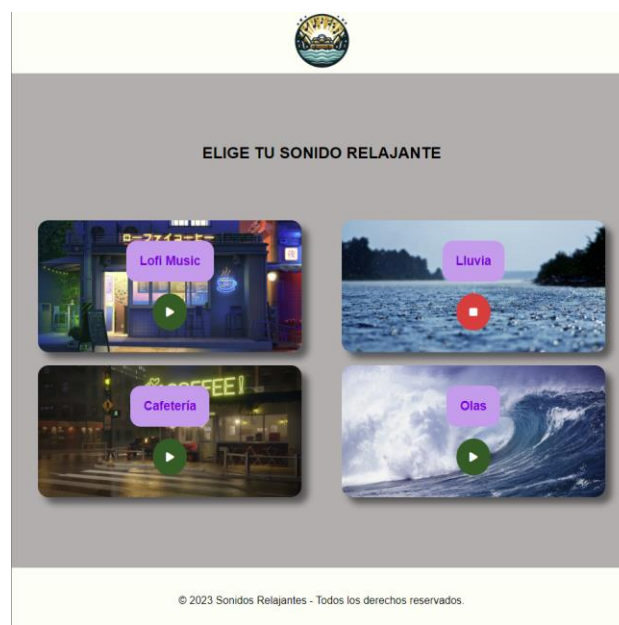
```
/**
 * Agrega un evento de escucha inicial a la sección de botones.
 */
1+ usages new *
function addEventoEscuchaInicial() {
    const contenedorBotones = document.getElementsByClassName('contenedor-grid')[0];
    contenedorBotones.addEventListener('click', toggleMusic);
}

window.addEventListener('DOMContentLoaded', addEventoEscuchaInicial);
```

De esta forma el usuario podrá seleccionar el sonido que prefiera cambiando el icono de un botón de "play" a un botón de "stop" y cambiando el fondo del icono de verde a rojo.



Llegados a este punto el usuario puede dejar la música elegida (se reproducirá de forma infinita) o elegir otra música:



En ese caso como se ve en la imagen el icono de la música previamente elegida vuelve a su estado por defecto y el botón de la música elegida es el que cambia de aspecto.



En el caso de que el usuario pulse sobre el icono de “stop”, la web reestablece todos los iconos y no produce ningún sonido:

