### Tailwind:

### SECTION Clases de Posicionamiento y Tamaño

* relative: Hace que un elemento se posicione de manera relativa respecto a su posición normal.
* absolute: Posiciona un elemento de forma absoluta respecto a su elemento contenedor más cercano con posición relativa.
* inset-0: Establece los valores de top, right, bottom y left a 0.
* top-20, top-8, top-5, top-0: Define la distancia del borde superior del contenedor al elemento.
* left-0: Establece el margen izquierdo del elemento en 0.
* h-full: Hace que el elemento tenga el mismo alto que su contenedor.
* w-full: Hace que el elemento ocupe todo el ancho de su contenedor.
* h-80: Define una altura fija de 20rem.
* sm:h-[75vh]: En pantallas pequeñas (sm:), establece el alto como el 75% de la altura de la ventana gráfica.
* h-64: Define una altura de 16rem.
* w-full: Hace que el ancho ocupe el 100% de su contenedor.

**Clases de Color**

* bg-black: Establece un fondo negro.
* bg-opacity-50: Define una opacidad del 50% para el color de fondo.
* text-white: Hace que el texto sea blanco.
* text-black: Hace que el texto sea negro.
* text-blue-600: Cambia el color del texto a un tono de azul.
* dark:text-blue-500: En temas oscuros, ajusta el texto a un azul más claro.

**Clases de Espaciado**

* mt-1, mt-28, mt-24: Añaden margen superior de diferentes tamaños.
* p-3, p-4, p-5: Añaden relleno interno al elemento.
* my-auto: Centra el elemento verticalmente usando márgenes automáticos.
* px-5, px-20: Añaden relleno horizontal (izquierda y derecha).
* pl-5: Añade relleno interno a la izquierda.
* mb-9, my-9: Añaden márgenes inferiores y verticales respectivamente.
* sm:px-5: Aplica relleno horizontal en pantallas pequeñas.

**Clases de Tipografía**

* font-bold: Aplica un peso de fuente fuerte.
* font-medium: Aplica un peso de fuente medio.
* text-lg: Define el tamaño del texto como grande (aproximadamente 1.125rem).
* text-xl, text-2xl, text-4xl, text-5xl, text-7xl, text-8xl, text-9xl: Define diferentes tamaños de texto.
* sm:text-5xl: Cambia el tamaño del texto a 5xl en pantallas pequeñas.
* leading-relaxed: Aumenta el espacio entre líneas.
* sm:leading-loose: Aumenta aún más el espacio entre líneas en pantallas pequeñas.
* underline: Subraya el texto.
* decoration-offset-4: Añade espacio entre el texto y la subrayado.

**Clases de Diseño Responsivo**

* sm:hidden: Oculta el elemento en pantallas pequeñas (sm).
* sm:grid: Convierte un contenedor en una cuadrícula en pantallas pequeñas.
* sm:grid-cols-3: Define 3 columnas en la cuadrícula en pantallas pequeñas.
* sm:gap-4: Establece un espacio entre los elementos de la cuadrícula en pantallas pequeñas.

**Clases de Imagen**

* object-cover: Hace que las imágenes llenen su contenedor recortándolas si es necesario.
* sm:object-center: Centra la imagen dentro de su contenedor en pantallas pequeñas.
* rounded: Aplica bordes redondeados a los elementos.
* hover:scale-105: Amplía el elemento al 105% de su tamaño original cuando se coloca el cursor sobre él.
* transition-transform: Añade una transición suave para los cambios de transformación.
* duration-300: Define una duración de 300ms para las transiciones.

**Clases de Flexbox**

* flex: Hace que el contenedor se comporte como un contenedor flex.
* justify-between: Distribuye los elementos dejando espacio igual entre ellos.
* justify-center: Centra los elementos horizontalmente.
* items-center: Centra los elementos verticalmente.
* hidden: Oculta el elemento.

**Clases de Botones y Decoración**

* bg-opacity-40: Ajusta la opacidad del fondo al 40%.
* rounded-full: Hace que los bordes sean completamente redondeados (circulares).
* shadow-xl: Aplica una sombra externa grande.
* hover:cursor-pointer: Cambia el cursor a una mano cuando se coloca sobre el elemento.  
    
    
  NAV

### ****Clases de Posicionamiento y Tamaño****

* relative: Establece que el elemento se posicionará de forma relativa a su posición normal.
* z-50: Asigna una prioridad de apilamiento muy alta al elemento (útil para que se muestre encima de otros elementos).
* w-full: Hace que el elemento ocupe todo el ancho disponible.
* h-20: Define una altura fija de 5rem para la imagen (80px).
* px-4: Agrega un relleno horizontal (izquierda y derecha) de 1rem.
* py-4: Agrega un relleno vertical (arriba y abajo) de 1rem.
* mt-3: Añade un margen superior de 0.75rem a los elementos.
* gap-2: Define un espacio de 0.5rem entre los elementos en un contenedor.
* sm:flex-row: Aplica el diseño de flexbox en fila para pantallas pequeñas (sm).
* sm:max-h-full: Permite que el contenedor tome el máximo alto en pantallas pequeñas.

### ****Clases de Fondo y Colores****

* bg-white: Establece un fondo blanco.
* bg-gray-200: Cambia el fondo a un tono claro de gris al hacer hover.
* hover:bg-gray-200: Cambia el fondo a gris claro al pasar el cursor.
* hover:bg-green-600: Cambia el fondo a un verde oscuro al pasar el cursor sobre el botón de iniciar sesión.
* hover:bg-red-600: Cambia el fondo a un rojo oscuro al pasar el cursor sobre el botón de cerrar sesión.
* bg-green-300: Aplica un fondo verde claro al botón de iniciar sesión.
* bg-red-300: Aplica un fondo rojo claro al botón de cerrar sesión.

### ****Clases de Tipografía****

* text-gray-700: Asigna un color gris oscuro al texto.
* text-gray-600: Asigna un color gris más claro al texto.
* hover:text-gray-900: Cambia el texto a un gris oscuro más intenso al pasar el cursor.
* fill-current: Hace que el color del ícono SVG sea el mismo que el texto actual.

### ****Clases de Espaciado y Bordes****

* rounded: Hace que los bordes de los elementos sean ligeramente redondeados.
* rounded hover:text-gray-900: Redondea los bordes de los botones y cambia el texto al gris oscuro al hacer hover.
* focus:outline-none: Elimina el contorno predeterminado al hacer foco en el botón.
* px-2: Añade un relleno horizontal pequeño (izquierda y derecha) de 0.5rem.
* py-2: Añade un relleno vertical pequeño (arriba y abajo) de 0.5rem.

### ****Clases de Flexbox****

* flex: Hace que el contenedor utilice flexbox.
* justify-between: Distribuye el contenido dejando espacio igual entre los elementos.
* items-center: Centra verticalmente los elementos dentro de un contenedor.
* flex-col: Hace que los elementos dentro del contenedor se dispongan en columna.
* sm:flex: Activa flexbox en pantallas pequeñas.

### ****Clases Responsivas****

* sm:hidden: Oculta el botón del menú en pantallas pequeñas.
* sm:block: Muestra un contenedor en pantallas pequeñas.
* sm:items-center: Centra los elementos verticalmente en pantallas pequeñas.
* sm:justify-between: Distribuye los elementos horizontalmente en pantallas pequeñas.

### ****Clases de Transición****

* opacity-0: Hace que el elemento sea completamente transparente.
* transition-all: Activa una transición suave para todos los cambios en las propiedades CSS.
* duration-1000: Define una duración de 1000ms (1 segundo) para las transiciones.
* ease-in-out: Define una transición suave al inicio y al final del cambio.

### ****Clases de Utilidad General****

* shadow-xl: Aplica una sombra grande al elemento.
* ring-1: Agrega un borde (ring) de 1px al elemento.
* ring-gray-900/5: Define el color del borde como un tono negro con opacidad del 5%.
* hidden: Oculta elementos.
* hover:cursor-pointer: Cambia el cursor a una mano al pasar sobre el elemento.

Modal, Inicio  
  
Aquí tienes un desglose detallado de las clases de Tailwind CSS utilizadas en este fragmento de HTML:

### ****Clases de Posicionamiento y Dimensiones****

* **fixed**: Fija el elemento en la pantalla (permanece en su posición incluso al hacer scroll).
* **inset-0**: Establece que el elemento ocupe todo el espacio del contenedor (top, right, bottom, left = 0).
* **absolute**: Posiciona el elemento de forma absoluta dentro de su contenedor relativo.
* **top-1/2**: Coloca el elemento en la mitad vertical del contenedor.
* **left-1/2**: Coloca el elemento en la mitad horizontal del contenedor.
* **transform**: Permite transformar el elemento (por ejemplo, moverlo, escalarlo, rotarlo, etc.).
* **-translate-x-1/2**: Mueve el elemento hacia la izquierda el 50% de su ancho.
* **-translate-y-1/2**: Mueve el elemento hacia arriba el 50% de su altura.
* **w-11/12**: Hace que el ancho sea el 11/12 del contenedor padre.
* **h-auto**: Permite que el alto del elemento se ajuste automáticamente según su contenido.
* **sm:h-[78vh]**: Para pantallas pequeñas, ajusta el alto al 78% del viewport.
* **sm:w-[40vw]**: Para pantallas pequeñas, ajusta el ancho al 40% del viewport.
* **sm:max-w-md**: Para pantallas pequeñas, establece un ancho máximo de 28rem (448px).

### ****Clases de Fondo y Colores****

* **bg-black**: Establece un fondo negro.
* **bg-opacity-50**: Aplica un 50% de opacidad al fondo.
* **bg-white**: Establece un fondo blanco.
* **bg-gray-50**: Establece un fondo gris muy claro (#F9FAFB).
* **from-gray-100** y **to-red-50**: Usados para un gradiente de fondo desde un gris claro hasta un rojo muy claro.
* **hover:bg-primary-700**: Cambia el fondo a un color oscuro específico al pasar el cursor.

### ****Clases de Bordes y Sombras****

* **rounded**: Hace que los bordes del elemento sean redondeados.
* **rounded-lg**: Aplica un redondeo más pronunciado en las esquinas.
* **shadow-xl**: Añade una sombra grande al elemento.
* **border**: Agrega un borde de 1px.
* **border-gray-300**: Establece el color del borde en un gris claro.

### ****Clases de Espaciado****

* **p-6**: Añade un relleno interno de 1.5rem alrededor del contenido.
* **px-9**: Añade un relleno horizontal de 2.25rem.
* **py-8**: Añade un relleno vertical de 2rem.
* **mb-6**: Añade un margen inferior de 1.5rem.
* **mx-auto**: Centra el elemento horizontalmente.

### ****Clases de Tipografía****

* **text-xl**: Define un tamaño de fuente grande (1.25rem o 20px).
* **font-bold**: Aplica un peso de fuente fuerte.
* **leading-tight**: Ajusta el espaciado entre líneas para que estén más juntas.
* **tracking-tight**: Reduce el espaciado entre caracteres.
* **text-sm**: Aplica un tamaño de fuente pequeño (0.875rem o 14px).
* **font-light**: Usa un peso de fuente ligero.
* **text-primary-600**: Establece un color personalizado (dependiendo del tema).
* **hover:underline**: Subraya el texto al pasar el cursor.

### ****Clases Flexbox****

* **flex**: Activa el modo flexbox para el contenedor.
* **flex-col**: Ordena los elementos en una columna.
* **items-center**: Centra los elementos hijos horizontalmente.
* **justify-center**: Centra los elementos hijos verticalmente.

### ****Clases de Transición y Hover****

* **focus:outline-none**: Elimina el contorno predeterminado al enfocar un elemento.
* **focus:ring-4**: Añade un contorno externo (ring) de 4px al elemento al enfocarlo.
* **focus:ring-primary-300**: Establece el color del contorno de enfoque en un tono específico.

### ****Clases para Formularios****

* **block**: Hace que el elemento sea un bloque.
* **w-full**: Hace que el elemento ocupe el 100% del ancho disponible.
* **p-2.5**: Añade un relleno de 0.625rem alrededor del input.
* **placeholder-gray-800**: Cambia el color del texto del marcador de posición a un gris oscuro.

### ****Clases Responsivas****

* **sm:flex**: Habilita flexbox en pantallas pequeñas.
* **sm:h-full**: Hace que el elemento ocupe todo el alto en pantallas pequeñas.
* **sm:w-full**: Hace que el elemento ocupe todo el ancho en pantallas pequeñas.
* **lg:py-0**: Elimina el relleno vertical en pantallas grandes.

# Utilidades Clave en el Código para el Slider y la Galería

## 1. **Funciones del Slider**

### ****1.1. showSlide(index)****

* **Propósito:** Muestra la diapositiva correspondiente al índice y oculta las demás.
* **Principales características:**
  + Utiliza clases CSS como active y hidden para controlar la visibilidad de las diapositivas.
  + Añade un efecto de aparición al texto dentro de cada diapositiva con un retraso de 300ms, utilizando style.opacity.

### ****1.2. next()****

* **Propósito:** Avanza a la siguiente diapositiva.
* **Lógica:**
  + Incrementa el índice de la diapositiva actual.
  + Usa el operador módulo (%) para garantizar que el índice vuelva a 0 después de la última diapositiva.

### ****1.3. prev()****

* **Propósito:** Retrocede a la diapositiva anterior.
* **Lógica:**
  + Decrementa el índice de la diapositiva actual.
  + Usa el operador módulo para manejar índices negativos y volver a la última diapositiva si se retrocede desde la primera.

### ****1.4. handleUserInteraction()****

* **Propósito:** Pausa temporalmente el auto-slider al detectar interacción del usuario y lo reinicia después de 5 segundos.
* **Lógica:**
  + Limpia el intervalo actual del auto-slider y reinicia el temporizador de reinicio.

### ****1.5. startAutoSlider()****

* **Propósito:** Activa el auto-slider si se especifica un intervalo (autoInterval).
* **Lógica:**
  + Usa setInterval para avanzar automáticamente al siguiente slide cada autoInterval milisegundos.

### ****1.6. Eventos de Control Manual****

* **Botón Adelante (next):**
  + Avanza al siguiente slide.
  + Pausa el auto-slider temporalmente.
* **Botón Atrás (prev):**
  + Retrocede al slide anterior.
  + Pausa el auto-slider temporalmente.

## 2. **Funciones de la Galería**

### ****2.1. modales\_galeria()****

* **Propósito:** Controla la apertura y cierre de modales en la galería al interactuar con las imágenes.
* **Principales características:**
  + **Clonación de Elementos:** Clona el elemento <li> clicado junto con su imagen y botón.
  + **Control de Modales:**
    - Muestra el modal cuando se hace clic en una imagen.
    - Cierra el modal al hacer clic fuera de la imagen o en el botón de cierre.
  + **Clases CSS Dinámicas:**
    - Ajusta las clases CSS del <li> y su <img> para adecuarse al diseño del modal.

### ****2.2. Eventos de Cierre****

* **Cerrar Modal:**
  + Usa closest() para detectar si el clic ocurrió en el botón de cierre.
  + Usa modal.contains(e.target) para determinar si el clic ocurrió fuera del modal.
  + Limpia el contenido del modal y lo oculta al cerrar.

## 3. **Elementos de JavaScript Esenciales**

### ****3.1. getElementById****

* **Uso:**
  + Obtiene referencias a elementos del DOM, como el slider, los botones de control y el contenedor de modales.
  + Ejemplo: const sliderElement = document.getElementById(sliderId);

### ****3.2. childElementCount****

* **Uso:**
  + Obtiene el número total de hijos directos de un elemento.
  + Ejemplo: const totalSlides = sliderElement.childElementCount;

### ****3.3. getElementsByTagName****

* **Uso:**
  + Recupera todos los elementos <li> dentro del slider y la galería.
  + Ejemplo: const slides = sliderElement.getElementsByTagName('li');

### ****3.4. classList****

* **Métodos Clave:**
  + add(): Agrega una clase a un elemento.
  + remove(): Elimina una clase de un elemento.
  + contains(): Verifica si un elemento tiene una clase específica.
* **Ejemplo:**
* element.classList.add('hidden');

element.classList.remove('active');

### ****3.5. setTimeout****

* **Uso:**
  + Retrasa la ejecución de una función por un tiempo específico.
  + Ejemplo: Retrasar la aparición del texto:
  + setTimeout(() => {
  + textContainer.style.opacity = "1";

}, 300);

### ****3.6. clearInterval y clearTimeout****

* **Uso:**
  + Detiene la ejecución de un intervalo o temporizador activo.
  + Ejemplo:
  + clearInterval(autoSlideInterval);

clearTimeout(userInteractionTimeout);

## 4. **Clases CSS Utilizadas en el Código**

### ****Slider****

* active: Muestra la diapositiva activa.
* hidden: Oculta las diapositivas inactivas.

### ****Galería****

* overflow-y-hidden: Oculta el scroll vertical cuando el modal está abierto.
* h-96: Define una altura fija para los elementos de la galería.
* sm:h-[45vh]: Ajusta la altura de las imágenes para pantallas pequeñas.

## 5. **Buenas Prácticas Identificadas**

* **Modularidad:** Cada función tiene una única responsabilidad, lo que facilita su mantenimiento.
* **Reutilización:** Funciones como showSlide() y handleUserInteraction() pueden ser llamadas desde diferentes contextos.
* **Interactividad:** Se manejan eventos de usuario con addEventListener para mejorar la experiencia.

# **CSS Dinámico:** Se ajustan las clases de los elementos dinámicamente para adaptarse al contexto. Documentación de Utilidades y Métodos en el Código JavaScript

Este documento describe las utilidades, métodos y funciones utilizadas en el código JavaScript proporcionado para la gestión de habitaciones y reservas.

## 1. **Funciones principales**

### ****1.1 dibujarHabitaciones(habitaciones)****

**Descripción:** Dibuja dinámicamente las habitaciones en un contenedor HTML utilizando los datos proporcionados.

**Parámetros:**

* habitaciones: Array de objetos con la información de cada habitación (nombre, descripción, precio, etc.).

**Funcionalidad:**

1. Itera sobre el array de habitaciones.
2. Genera una estructura HTML para cada habitación.
3. Inserta el HTML generado en el contenedor correspondiente (contenedor\_habitaciones).

**Uso:** Llamado al cargar las habitaciones o filtrar las disponibles.

### ****1.2 mostrarHabitaciones()****

**Descripción:** Obtiene la lista de habitaciones desde una API REST y las dibuja utilizando dibujarHabitaciones.

**Flujo:**

1. Realiza una solicitud fetch a la URL de la API.
2. Convierte la respuesta a JSON.
3. Llama a dibujarHabitaciones con los datos obtenidos.
4. Muestra mensajes de error en caso de fallo.

**Uso:** Se ejecuta al cargar la página habitaciones.html.

### ****1.3 reservas()****

**Descripción:** Configura restricciones y eventos para seleccionar fechas de reserva.

**Funcionalidad:**

1. Asigna fechas mínimas iniciales para los campos de inicio y fin de reserva.
2. Actualiza dinámicamente la fecha mínima de finalización al cambiar la fecha de inicio.

**Uso:** Garantiza que las fechas seleccionadas sean válidas y coherentes.

### ****1.4 mostrarDisponibilidad()****

**Descripción:** Filtra habitaciones disponibles según los criterios del formulario (capacidad y fechas).

**Flujo:**

1. Escucha el evento submit del formulario.
2. Obtiene los valores de los campos del formulario.
3. Filtra habitaciones disponibles según la capacidad máxima.
4. Muestra las habitaciones disponibles o un mensaje de no disponibilidad.

**Uso:** Llamado cuando el usuario busca habitaciones disponibles.

### ****1.5 setupReservaModal()****

**Descripción:** Administra la apertura y cierre del modal para gestionar reservas.

**Funcionalidad:**

1. Configura eventos de clic para botones de reserva y cancelación.
2. Muestra el modal con información de la habitación seleccionada.
3. Envía una solicitud PUT para confirmar o cancelar reservas.
4. Actualiza dinámicamente el estado de la interfaz tras la reserva o cancelación.

**Uso:** Permite al usuario interactuar con habitaciones específicas.

## 2. **Utilidades adicionales**

### ****2.1 Manejo de Fechas****

* **new Date()**: Crea instancias de fecha.
* **toISOString().split("T")[0]**: Obtiene la fecha en formato YYYY-MM-DD.
* **setDate()**: Ajusta el día del mes para calcular fechas mínimas dinámicamente.

### ****2.2 Métodos de Fetch****

* **GET:**
* fetch('URL')
* .then(response => response.json())

.then(data => console.log(data));

Utilizado en mostrarHabitaciones y mostrarDisponibilidad para obtener datos.

* **PUT:**
* fetch('URL', {
* method: 'PUT',
* headers: { 'Content-Type': 'application/json' },
* body: JSON.stringify(datosActualizados)
* })
* .then(response => response.json())

.then(data => console.log(data));

Utilizado en setupReservaModal para actualizar reservas.

## 3. **Componentes HTML Generados Dinámicamente**

### ****Habitación****

Estructura HTML generada por dibujarHabitaciones:

<div class="bg-white rounded overflow-hidden">

<img src="{imagen}" alt="Imagen de la habitación" class="w-full h-52 object-cover" />

<div class="p-6">

<h3 class="text-lg font-bold">{nombre}</h3>

<p>{descripcion}</p>

<p>Wifi: {wifi}</p>

<p>Max Personas: {max\_personas}</p>

<p>Precio: {precio}</p>

<a class="btn-reservar">Reservar</a>

</div>

</div>

## 4. **Manejo de Errores**

* **Try-Catch:** Utilizado en mostrarDisponibilidad y setupReservaModal para capturar errores durante las solicitudes fetch.
* **Mensajes de Consola:**
* console.warn("Elemento no encontrado");

console.error("Error:", error);

Proporciona información detallada durante el desarrollo.

## 5. **Interacciones del Usuario**

### ****Botones de Reserva****

* **Clase:** btn-reservar.
* **Función:** Abre el modal para gestionar reservas.

### ****Botones de Cancelación****

* **Clase:** btn-cancelar.
* **Función:** Cancela una reserva y actualiza el estado de la habitación.

## 6. **Dependencias del Proyecto**

* **API REST:** URL: https://apiproject-1c5o.onrender.com/habitaciones
* **Session Storage:**

sessionStorage.getItem("usuario\_activo")

Almacena y recupera el usuario activo para asociar reservas.

## 7. **Eventos Escuchados**

### Formularios

* submit: Capturado en mostrarDisponibilidad y el modal de reservas.

### Campos de Fecha

* change: Actualiza las restricciones de selección de fechas.

### Botones

* click: Gestiona las reservas y el cierre del modal.