**Guía del Laboratorio - Sitio Web Avanzado con HTML, CSS, JS y DOM**

# **1. Introducción**

Este laboratorio implementa un sitio web moderno que incluye navegación dinámica, carrusel automático de imágenes, contenido de tipo periódico, formulario de contacto y gráficos interactivos usando Chart.js. La interacción se realiza mediante el DOM.

# **2. Estructura del Proyecto**

* index.html – Estructura principal del sitio
* style.css – Archivo de estilos CSS
* script.js – Código JavaScript con manipulación DOM
* img1.jpg, img2.jpg, img3.jpg – Imágenes para el carrusel

# **3. index.html - Contenido**

• Header: contiene el título del sitio y el menú de navegación.

• Secciones: cada sección tiene un id para ser mostrada u ocultada por JavaScript (DOM).

• Carrusel: se visualiza en la sección 'home' y rota cada 5 segundos.

• Información: organizada en 3 columnas tipo periódico.

• Estadística: incluye un gráfico dinámico con filtro desplegable.

• Formulario de contacto: validado y gestionado con JavaScript.

# **4. style.css - Estilo Visual**

* visible: clase aplicada dinámicamente para mostrar secciones.
* .grid-noticias: utiliza CSS Grid para imitar la apariencia de un periódico.
* carrusel img: define tamaño adaptable para las imágenes.
* Estilos de formularios, botones, cabecera y pie de página.

# **5. script.js - JavaScript con DOM**

* Carrusel de imágenes: cambia cada 5 segundos con setInterval.
* mostrarSeccion(id): usa querySelectorAll y classList para mostrar u ocultar secciones dinámicamente.
* enviarFormulario(): valida que todos los campos estén completos y muestra mensaje de éxito o error.
* actualizarGrafico(): actualiza los datos del gráfico según la categoría seleccionada (ventas o usuarios).

# 6. Gráficos con Chart.js

• Chart.js es una librería externa para generar gráficos en canvas.

• Se incluye desde CDN: https://cdn.jsdelivr.net/npm/chart.js

• Se crea el gráfico usando new Chart(contexto, configuración).

# 7. ¿Qué es el DOM?

El DOM (Document Object Model) es una estructura de árbol jerárquica que representa el contenido HTML de una página web. Permite acceder, modificar y gestionar los elementos del documento HTML de forma dinámica con JavaScript.

# 8. Comandos DOM Usados

* document.getElementById(): obtiene un elemento HTML por su ID.
* document.querySelectorAll(): selecciona múltiples elementos con un selector CSS.
* element.classList.add()/remove(): agrega o remueve clases CSS.
* setInterval(): ejecuta repetidamente una función cada cierto tiempo.
* value, .textContent: accede o modifica valores y contenido.