|  |
| --- |
| prueba técnica becario ESPOT  Desarrollo de visualizador de productos |
| Imagen que contiene dibujo, plato  Descripción generada automáticamente |



# ESPOT

# Asesoría Avanzada en Soluciones y Desarrollos Digitales

Morelos 53, Del Carmen, Coyoacán, 04100 Ciudad de México, CDMX.

hola@espot-mkt.com

**Nombre del Proyecto:** Venta de Números

**Autor:** Eugenia Ibarra M.

**Fecha:** 19/Jun/2024

**Versión del Documento:** 1.0

#### Tabla de Contenidos

1. Introducción
2. Descripción General
3. Tecnologías Utilizadas
4. Diseño de la Solución
   * 4.1 HTML
   * 4.2 CSS
   * 4.3 JavaScript
5. Conclusión
6. Referencias

#### Introducción

Este documento técnico describe el diseño de la solución para una página web interactiva dedicada a la venta de productos numéricos.

La solución utiliza HTML, CSS y JavaScript para estructurar, estilizar y añadir interactividad a la página.

### **Descripción General**

#### La solución al problema, para este proyecto, consiste en una página web que permite a los usuarios filtrar productos numéricos por categorías y reiniciar los filtros para ver todos los productos disponibles. Esta funcionalidad mejora la experiencia del usuario al permitirle buscar y seleccionar productos de manera eficiente.

### **Tecnologías Utilizadas**

**HTML (HyperText Markup Language)**: Para la estructura básica del contenido de la página web.

**CSS (Cascading Style Sheets)**: Para estilizar los elementos HTML y mejorar la apariencia visual.

**JavaScript**: Para añadir interactividad, permitiendo la aplicación de filtros dinámicos y el reinicio de estos.

### **Diseño de la Solución**

#### 4.1 HTML

El archivo HTML define la estructura de la página, incluyendo el título principal, los filtros, la lista de productos y un botón para resetear los filtros.

html

Copiar código

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Venta de números</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<!-- Contenedor principal de la página -->

<div class="container">

<h1>Por ocasión especial, ¡puede adquirir números!</h1>

<!-- Sección de filtros -->

<div id="filters">

<h2>Filtros:</h2>

<label><input type="checkbox" class="filter" data-filter="categoria1"> 1</label>

<label><input type="checkbox" class="filter" data-filter="categoria2"> 2</label>

<label><input type="checkbox" class="filter" data-filter="categoria3"> 3</label>

<label><input type="checkbox" class="filter" data-filter="categoria4"> 4</label>

</div>

<!-- Lista de productos -->

<div id="product-list">

<div class="product" data-categoria="categoria1">

<h3>Producto 1</h3>

<p>Usted puede llevarse el hermoso número 1.</p>

</div>

<div class="product" data-categoria="categoria2">

<h3>Producto 2</h3>

<p>Usted puede llevarse el maravilloso número 2.</p>

</div>

<div class="product" data-categoria="categoria3">

<h3>Producto 3</h3>

<p>Usted puede llevarse el inigualable número 3.</p>

</div>

<div class="product" data-categoria="categoria4">

<h3>Producto 4</h3>

<p>Usted puede llevarse el único número 4.</p>

</div>

</div>

<!-- Botón para resetear los filtros -->

<button id="reset-button">Limpiar Filtros</button>

</div>

<script src="script.js"></script>

</body>

</html>

#### 4.2 CSS

El archivo CSS aplica estilos a los elementos HTML, mejorando la apariencia visual de la página web.

css

Copiar código

/\* Estilo para el cuerpo de la página \*/

body {

font-family: Arial, sans-serif; /\* Establece la fuente del texto \*/

line-height: 1.6; /\* Establece la altura de línea \*/

background-color: #f0f0f0; /\* Establece el color de fondo \*/

margin: 0; /\* Elimina el margen por defecto \*/

padding: 0; /\* Elimina el relleno por defecto \*/

}

/\* Estilo para el contenedor principal \*/

.container {

width: 80%; /\* Establece el ancho del contenedor \*/

margin: auto; /\* Centra el contenedor horizontalmente \*/

background-color: #90cad1; /\* Establece el color de fondo \*/

padding: 20px; /\* Añade relleno \*/

box-shadow: 0 0 10px rgba(35, 163, 202, 0.32); /\* Añade sombra \*/

}

/\* Estilo para los encabezados h1, h2, h3 \*/

h1, h2, h3 {

color: #1433e6; /\* Establece el color del texto \*/

}

/\* Estilo para el contenedor de filtros \*/

#filters {

margin-bottom: 20px; /\* Añade margen inferior \*/

}

/\* Estilo para cada producto \*/

.product {

border: 1px solid #4d78e5; /\* Establece el borde \*/

background-color: #f9f9f9; /\* Establece el color de fondo \*/

padding: 10px; /\* Añade relleno \*/

margin-bottom: 10px; /\* Añade margen inferior \*/

}

/\* Estilo para los encabezados h3 dentro de productos \*/

.product h3 {

margin-top: 0; /\* Elimina el margen superior \*/

}

#### 4.3 JavaScript

#### El archivo JavaScript añade funcionalidad dinámica a la página, permitiendo a los usuarios aplicar y resetear filtros.

javascript

Copiar código

// Espera a que el DOM esté completamente cargado antes de ejecutar el script

document.addEventListener('DOMContentLoaded', function() {

// Selecciona todos los elementos con la clase 'filter' y los almacena en 'filters'

const filters = document.querySelectorAll('.filter');

// Selecciona todos los elementos con la clase 'product' y los almacena en 'products'

const products = document.querySelectorAll('.product');

// Selecciona el botón con el id 'reset-button' y lo almacena en 'resetButton'

const resetButton = document.getElementById('reset-button');

// Función para aplicar los filtros seleccionados

function applyFilters() {

// Añade un event listener a cada filtro

filters.forEach(function(filter) {

filter.addEventListener('change', function() {

// Array para almacenar los filtros seleccionados

let selectedFilters = [];

// Recorre todos los filtros y añade los seleccionados al array 'selectedFilters'

filters.forEach(function(f) {

if (f.checked) {

selectedFilters.push(f.getAttribute('data-filter'));

}

});

// Recorre todos los productos y muestra/oculta según los filtros seleccionados

products.forEach(function(product) {

// Obtiene la categoría del producto

let productCategory = product.getAttribute('data-categoria');

// Variable para determinar si el producto es visible

let isVisible = false;

// Comprueba si el producto debe ser visible según los filtros seleccionados

selectedFilters.forEach(function(selectedFilter) {

if (selectedFilter === 'todos' || selectedFilter === productCategory) {

isVisible = true;

}

});

// Muestra u oculta el producto según el valor de 'isVisible'

if (isVisible) {

product.style.display = 'block';

} else {

product.style.display = 'none';

}

});

});

});

}

// Función para resetear los filtros y mostrar todos los productos

function resetFilters() {

// Desmarca todos los filtros

filters.forEach(function(filter) {

filter.checked = false;

});

// Muestra todos los productos

products.forEach(function(product) {

product.style.display = 'block';

});

}

// Event listener para el botón de resetear filtros

resetButton.addEventListener('click', function() {

resetFilters();

});

// Aplicar filtros por defecto al cargar la página

applyFilters();

});

#### Conclusión

La integración de HTML, CSS y JavaScript en esta actividad permite crear una página web interactiva y atractiva que facilita la búsqueda y selección de productos por parte de los usuarios. La estructura clara, el diseño estilizado y la funcionalidad dinámica proporcionan una experiencia de usuario mejorada.

### **Referencias**

Array.prototype.filter() - JavaScript | MDN. (2024). MDN

Web Docs.

margin-bottom - CSS | MDN. (2024). MDN Web Docs. https://developer.mozilla.org/es/docs/Web/CSS/margin-bottom

# W3Schools.com. (2024). HTML class Attribute