<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

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

<title>Prisma Cuadrangular 3D Dinámico</title>

<style>

body {

background-color: #003366;

color: white;

font-family: Arial, sans-serif;

font-size: 18px;

text-align: center;

}

input, button {

margin: 10px;

padding: 10px;

font-size: 18px;

}

.resultados {

margin-top: 20px;

}

.tabla-resultados {

width: 50%;

margin: 0 auto;

background-color: #002244;

color: white;

border-collapse: collapse;

font-size: 18px;

}

.tabla-resultados th, .tabla-resultados td {

padding: 10px;

border: 1px solid white;

}

#cerrarPrograma {

background-image: linear-gradient(to right, red, yellow, green, blue, violet);

color: black;

font-weight: bold;

cursor: pointer;

border: none;

padding: 10px;

margin-top: 20px;

}

#prisma3D {

width: 100%;

height: 500px;

}

</style>

</head>

<body>

<h1>Prisma Cuadrangular 3D Dinámico</h1>

<div>

<label for="altura">Altura (cm): </label>

<input type="number" id="altura" min="1" placeholder="Altura en cm" required>

<label for="ancho">Ancho (cm): </label>

<input type="number" id="ancho" min="1" placeholder="Ancho en cm" required>

<label for="profundidad">Profundidad (cm): </label>

<input type="number" id="profundidad" min="1" placeholder="Profundidad en cm" required>

<button onclick="crearPrisma3D()">Calcular y Dibujar</button>

</div>

<div id="prisma3D"></div>

<div class="resultados">

<h2>Resultados</h2>

<table class="tabla-resultados">

<tr>

<th>Área Total (cm²)</th>

<th>Volumen (cm³)</th>

</tr>

<tr>

<td id="areaTotal">0</td>

<td id="volumen">0</td>

</tr>

</table>

</div>

<button id="cerrarPrograma" onclick="cerrarPrograma()">Cerrar Programa</button>

<script src="https://cdnjs.cloudflare.com/ajax/libs/three.js/r128/three.min.js"></script>

<script>

let scene, camera, renderer, prisma;

function getRandomColor() {

const letters = '0123456789ABCDEF';

let color = '#';

for (let i = 0; i < 6; i++) {

color += letters[Math.floor(Math.random() \* 16)];

}

return color;

}

function crearPrisma3D() {

const altura = parseFloat(document.getElementById('altura').value);

const ancho = parseFloat(document.getElementById('ancho').value);

const profundidad = parseFloat(document.getElementById('profundidad').value);

if (altura > 0 && ancho > 0 && profundidad > 0) {

const areaLateral1 = altura \* ancho;

const areaLateral2 = altura \* profundidad;

const areaBase = ancho \* profundidad;

const areaTotal = 2 \* (areaLateral1 + areaLateral2 + areaBase);

const volumen = altura \* ancho \* profundidad;

document.getElementById('areaTotal').textContent = areaTotal.toFixed(2);

document.getElementById('volumen').textContent = volumen.toFixed(2);

// Crear el prisma 3D

iniciarEscena(ancho, altura, profundidad);

} else {

alert("Por favor, ingresa valores válidos.");

}

}

function iniciarEscena(ancho, altura, profundidad) {

// Eliminar la escena previa si existe

if (scene) {

while (scene.children.length > 0) {

scene.remove(scene.children[0]);

}

}

scene = new THREE.Scene();

camera = new THREE.PerspectiveCamera(75, window.innerWidth / 500, 0.1, 1000);

renderer = new THREE.WebGLRenderer();

renderer.setSize(window.innerWidth, 500);

document.getElementById('prisma3D').appendChild(renderer.domElement);

// Prisma geometría

const geometry = new THREE.BoxGeometry(ancho, altura, profundidad);

// Material multicolor aleatorio

const materials = [

new THREE.MeshBasicMaterial({ color: getRandomColor(), side: THREE.DoubleSide }),

new THREE.MeshBasicMaterial({ color: getRandomColor(), side: THREE.DoubleSide }),

new THREE.MeshBasicMaterial({ color: getRandomColor(), side: THREE.DoubleSide }),

new THREE.MeshBasicMaterial({ color: getRandomColor(), side: THREE.DoubleSide }),

new THREE.MeshBasicMaterial({ color: getRandomColor(), side: THREE.DoubleSide }),

new THREE.MeshBasicMaterial({ color: getRandomColor(), side: THREE.DoubleSide })

];

prisma = new THREE.Mesh(geometry, materials);

scene.add(prisma);

camera.position.z = 100;

// Iniciar la animación de rotación

animar();

}

function animar() {

requestAnimationFrame(animar);

// Rotación del prisma

prisma.rotation.x += 0.01;

prisma.rotation.y += 0.01;

renderer.render(scene, camera);

}

function cerrarPrograma() {

window.close();

}

</script>

</body>

</html>