**Estrutura HTML:**

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Galeria de Imagens</title>

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

</head>

<body>

<div class="gallery">

<div class="filter">

<button onclick="filterImages('all')">Todas</button>

<button onclick="filterImages('natureza')">Natureza</button>

<button onclick="filterImages('cidade')">Cidade</button>

<button onclick="filterImages('tecnologia')">Tecnologia</button>

</div>

<div class="images">

<div class="image natureza"><img src="natureza1.jpg" alt="Natureza 1"></div>

<div class="image cidade"><img src="cidade1.jpg" alt="Cidade 1"></div>

<div class="image tecnologia"><img src="tecnologia1.jpg" alt="Tecnologia 1"></div>

<!-- Adicione mais imagens conforme necessário -->

</div>

</div>

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

</body>

</html>

* <!DOCTYPE html>: Define que o documento é do tipo HTML5.
* <html lang="en">: Define a linguagem do documento como inglês (pode ser alterada para pt-BR se preferir).
* <head>: Contém metadados do documento, como o título da página e a ligação para o arquivo de estilo CSS (styles.css).
* <body>: Contém o conteúdo visível da página, incluindo a galeria de imagens e o script JavaScript (script.js).

A galeria de imagens é organizada em duas seções principais: a seção de filtro, com botões para selecionar categorias, e a seção de imagens, onde as imagens são exibidas.

**2. Estilos CSS (styles.css):**

css

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

.gallery {

display: flex;

flex-direction: column;

align-items: center;

}

.filter {

margin-bottom: 20px;

}

.filter button {

margin-right: 10px;

padding: 5px 10px;

background-color: #3498db;

color: #fff;

border: none;

cursor: pointer;

}

.images {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

gap: 10px;

}

.image {

overflow: hidden;

border-radius: 8px;

box-shadow: 0 0 5px rgba(0, 0, 0, 0.3);

}

.image img {

width: 100%;

height: auto;

transition: transform 0.3s ease-in-out;

}

.image:hover img {

transform: scale(1.1);

}

* Define estilos para o corpo da página, como a família de fontes, margens e preenchimentos.
* Estiliza a galeria (gallery) com um layout de coluna (flex-direction: column) e alinha os elementos ao centro (align-items: center).
* Estiliza os botões de filtro (filter button) com cores de fundo, texto e bordas, além de adicionar um cursor de ponteiro ao passar o mouse sobre eles.
* Define a aparência da grade de imagens (images) usando o modelo de grade CSS (display: grid) e especificando o número de colunas e espaçamento entre os elementos.
* Estiliza cada imagem (image) com bordas arredondadas, sombra e uma transição suave de escala ao passar o mouse sobre elas.

**3. JavaScript (script.js):**

javascript

function filterImages(category) {

const images = document.querySelectorAll('.image');

images.forEach(image => {

if (category === 'all') {

image.style.display = 'block';

} else if (image.classList.contains(category)) {

image.style.display = 'block';

} else {

image.style.display = 'none';

}

});

}

* Define uma função filterImages(category) que é chamada quando um botão de filtro é clicado.
* Obtém todas as imagens da galeria usando document.querySelectorAll('.image').
* Para cada imagem, verifica se ela pertence à categoria selecionada ('all' para todas as imagens ou uma categoria específica como 'natureza', 'cidade', 'tecnologia', etc.).
* Define a exibição da imagem para block se ela corresponder à categoria selecionada ou para none se não corresponder, ocultando assim as imagens não selecionadas.

Esse código cria uma galeria de imagens interativa e responsiva, permitindo que os usuários filtrem as imagens por categoria. As imagens são exibidas com animações suaves ao passar o mouse sobre elas, criando uma experiência interativa para o usuário.