

**RELATÓRIO DE AULA PRÁTICA**

**Curso:** Engenharia de Software

**Matéria:** Desenvolvimento de JavaScript

**Aluno**: Marcelo Silva de Almeida

**Data:** 21/05/2025

**Assunto:** Página web com animação utilizando o elemento canvas

1 – Código HTML

<!DOCTYPE html>

<html>

<head>

  <title>Página web com animação utilizando o elemento <canvas></title>

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

</head>

<body>

  <canvas id="canvas" width="400" height="400"></canvas>

  <script src="script.js"></script>

</body>

</html>

2 – Código CSS

 #canvas {

      border: 1px solid black;

    }

3 – Código JavaScript

const canvas = document.getElementById('canvas');

const ctx = canvas.getContext('2d');

//  tamanho do canvas

canvas.width = 400;

canvas.height = 400;

//  array de círculos

let circles = [];

for (let i = 0; i < 10; i++) {

  circles.push({

    x: Math.random() \* canvas.width,

    y: Math.random() \* canvas.height,

    radius: Math.random() \* 20 + 10,

    speedX: Math.random() \* 2 - 1,

    speedY: Math.random() \* 2 - 1,

    color: getRandomColor()

  });

}

//  cor aleatória

function getRandomColor() {

  let r = Math.floor(Math.random() \* 256);

  let g = Math.floor(Math.random() \* 256);

  let b = Math.floor(Math.random() \* 256);

  return `rgb(${r}, ${g}, ${b})`;

}

//  animação

function animate() {

  // Limpar

  ctx.clearRect(0, 0, canvas.width, canvas.height);

  // círculos

  for (let i = 0; i < circles.length; i++) {

    let circle = circles[i];

    ctx.beginPath();

    ctx.arc(circle.x, circle.y, circle.radius, 0, 2 \* Math.PI);

    ctx.fillStyle = circle.color;

    ctx.fill();

    // Atualizar a posição

    circle.x += circle.speedX;

    circle.y += circle.speedY;

    // Verificar colisão com as bordas do canvas

    if (circle.x + circle.radius > canvas.width || circle.x - circle.radius < 0) {

      circle.speedX = -circle.speedX;

      circle.color = getRandomColor();

    }

    if (circle.y + circle.radius > canvas.height || circle.y - circle.radius < 0) {

      circle.speedY = -circle.speedY;

      circle.color = getRandomColor();

    }

  }

  //próximo frame de animação

  requestAnimationFrame(animate);

}

// Iniciar a animação

animate();

4 – Imagens com resultados



