

# DESENHO TÉCNICO E COMPUTAÇÃO GRÁFICA

Ms Márcio J Morais

**GRANTiETÉ**  
F A C U L D A D E

Engenharia da Computação

# **FACULDADE GRAN TIETÊ**

Av. 15 de Novembro, 125 - Centro - Barra Bonita/SP - CEP: 17340-047

(14) 3642-3219

[contato@grantiete.com.br](mailto:contato@grantiete.com.br)

<https://grantiete.com.br/>

## **Professor Mestre Márcio Jesus de Moraes**

[marciojmoraes@gmail.com](mailto:marciojmoraes@gmail.com)

## **Bacharelado em Engenharia Agrônômica**

### **Desenho Técnico e Expressão Gráfica**

**1.º Bimestre – 04/08 a 09/04**

**2.º Bimestre – 10/04 a 27/06**

### **Turmas/Termos**

Eng. Comp.: 1.º, 2.º, 3.º; 4.º, 5.º, 6.º, 8.º, 9.º e 10.º

Eng. Civil: 4.º, 5.º e 6.º

**Terças-feiras - 19h00 – 22h30 - Sala Allan Turing**

# AULA 01 - Fundamentos do Canvas HTML5

## 1. Objetivo da Aula

- Compreender o que é o elemento Canvas
- Entender o conceito de contexto 2D
- Dominar o sistema de coordenadas do Canvas
- Criar sua primeira área de desenho programática

## 2. O que é o Canvas?

O Canvas é um elemento HTML5 que funciona como uma "tela em branco" onde podemos desenhar programaticamente usando JavaScript. Pense nele como uma folha de papel digital onde, ao invés de usar lápis ou caneta, usamos código para criar formas, linhas e desenhos.

Características Principais:

- Bitmap: Trabalha com pixels (diferente do SVG que é vetorial);
- Programático: Tudo é desenhado via JavaScript;
- Versátil: Pode criar desde gráficos simples até jogos complexos;
- Tempo Real: Permite animações e interatividade.

### Elemento Canvas

Sintaxe Básica HTML:

```
html
<canvas id="meuCanvas" width="800" height="600">
  Seu navegador não suporta Canvas
</canvas>
```

### Propriedades Importantes:

**width:** Largura em pixels (padrão: 300px)

**height:** Altura em pixels (padrão: 150px)

**id:** Identificador para acessar via JavaScript

### Diferença Importante:

html

```
<!-- CORRETO: Dimensões no HTML -->  
<canvas width="800" height="600"></canvas>  
  
<!-- INCORRETO: Dimensões no CSS causam distorção -->  
<canvas style="width: 800px; height: 600px;"></canvas>
```

## Contexto 2D

O contexto é a "interface de desenho" do Canvas. É através dele que executamos todos os comandos de desenho.

Como Obter o Contexto:

javascript

```
// 1. Pegar referência do canvas  
const canvas = document.getElementById('meuCanvas');  
  
// 2. Obter o contexto 2D  
const ctx = canvas.getContext('2d');  
  
// Agora 'ctx' é nossa "caneta digital"
```

## Tipos de Contexto:

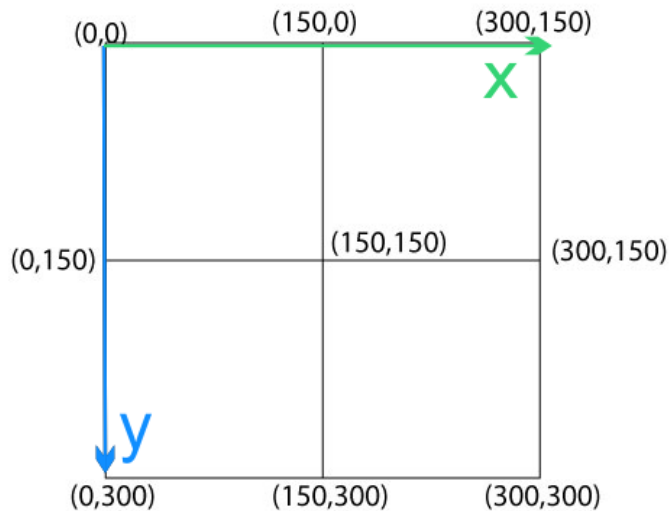
- '2d': Para desenhos 2D (nossa aula)
- 'webgl': Para gráficos 3D avançados

## Sistema de Coordenadas

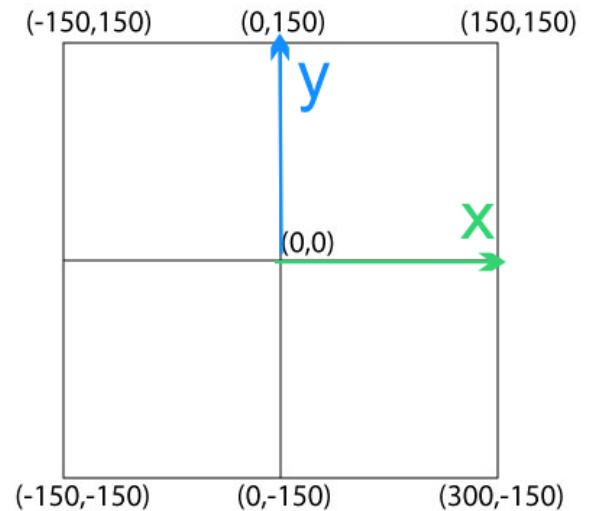
O Canvas usa um sistema de coordenadas diferente do matemático tradicional:

Características do Sistema:

# Canvas



# Cartesiano



## Pontos Importantes:

- Origem (0,0): Canto superior esquerdo
- Eixo X: Cresce para a direita
- Eixo Y: Cresce para baixo (invertido!)
- Unidade: Pixels

## Exemplos de Coordenadas:

```
javascript

// Canvas de 800x600 pixels
const canvas = document.getElementById('canvas');
canvas.width = 800;
canvas.height = 600;

// Pontos importantes:
// Centro: (400, 300)
// Canto inferior direito: (800, 600)
// Meio da borda superior: (400, 0)
```

## Exemplo Prático

Vamos criar nossa primeira área de desenho:

HTML:

```

html

<!DOCTYPE html>
<html>
<head>
  <title>Minha Primeira Área de Desenho</title>
</head>
<body>
  <h1>Computação Gráfica - Aula 1</h1>

  <!-- Nossa tela de desenho -->
  <canvas id="areaDesenho" width="800" height="600"
    style="border: 2px solid #333;">
    Seu navegador não suporta Canvas
  </canvas>

  <!-- Informações da tela -->
  <p>Dimensões: 800 x 600 pixels</p>

  <script>
    // Obter referência do canvas
    const canvas = document.getElementById('areaDesenho');
    const ctx = canvas.getContext('2d');

    // Confirmar dimensões
    console.log('Largura:', canvas.width);
    console.log('Altura:', canvas.height);
    console.log('Contexto obtido:', ctx);
  </script>
</body>
</html>

```

## Conceitos Importantes

### 1. Diferença entre Canvas e Outros Elementos

**DIV:** Contêiner para outros elementos

**IMG:** Exibe imagem estática

**CANVAS:** Área para desenho programático

### 2. Por que JavaScript?

- HTML define a estrutura (o canvas vazio)

- CSS define a aparência (bordas, posição)
- JavaScript define o conteúdo (o que é desenhado)

### **3. Analogia com o Mundo Real**

- Canvas = Tela em branco
- Contexto 2D = Conjunto de pincéis e ferramentas
- JavaScript = Suas mãos controlando as ferramentas