# Offset

As classes utilitárias de offsets no **WPF** permitem ajustar a posição de elementos com as propriedades CSS top, right, bottom e left. Elas são projetadas para serem intuitivas, seguindo uma sintaxe clara e flexível, compatível com unidades como `%`, `px`, `dvh` e `dvw`.

## Como Funcionam as Classes de Offsets

As classes de offsets seguem o padrão: (propriedade)(valor)(unidade), onde:

- **Propriedade**: `top`, `right`, `bottom` ou `left`.
- **Valor**: Um número inteiro (positivo ou negativo) que define o deslocamento.
- **Unidade**: Opcional, pode ser `%`, `px`, `dvh` (altura dinâmica da viewport) e `dvw` (largura dinâmica da viewport). Se omitida, o valor é tratado como unitless (ex.: `top(10)` equivale a `top: 10;`).

Isso significa que uma classe válida segue este formato:

```
<div class="fixed top(50%) left(20px)">
  <!-- Esse elemento html definiu os seguintes atributos css - top: 50%; left: 20px; -->
</div>

<div class="absolute right(-10dvw) bottom(100)">
  <!-- Esse elemento html definiu os seguintes atributos css - right: -10dvw; bottom: 100; -->
</div>
```

## Usando Funções de Classe para Centralização

As classes utilitárias de centralização do **WPF** simplificam o alinhamento de elementos horizontalmente, verticalmente ou em ambos os eixos. Elas combinam as propriedades left, top e transform para posicionar elementos com precisão, especialmente em elementos com position: `absolute`, `fixed` ou `sticky`.

### Como Funcionam as Classes de Centralização

As classes de centralização seguem uma sintaxe simples e intuitiva: **center(tipo)**, onde o tipo define o eixo de centralização:

As classes de centralização **são ideais para elementos com position:** `absolute`, `fixed` ou `sticky`, **dentro ou fora de um contêiner pai com position semelhante**. Certifique-se de que o elemento pai tenha dimensões definidas de alguma maneira (ex.: width e height) para que a centralização funcione corretamente.

| Classe Utilitária | Comportamento |
|---|---|
| `center(h)` | *Centraliza horizontalmente o elemento.* |
| `center(v)` | *Centraliza verticalmente o elemento.* |
| `center(c)` | *Centraliza em **ambos** os eixos (horizontal e vertical).* |

Com essa sintaxe, confira alguns exemplos dessas funções de classe em execução:

```
<div class="fixed top(20px) center(h)">
  <!-- No seguinte elemento html, a seguinte configuração css foi atribuída - top: 20px; left: 50%; transform: translateX(-50%); -->
</div>

<div class="absolute top(5%) center(v)">
  <!-- No seguinte elemento html, a seguinte configuração css foi atribuída - left: 5%; top: 50%; transform: translateY(-50%); -->
</div>

<div class="relative center(c)">
  <!-- No seguinte elemento html, a seguinte configuração css foi atribuída - left: 50%; top: 50%; transform: translate(-50%, -50%); -->
</div>
```

# Notas

> **Posicionamento**: As classes de centralização requerem que o elemento tenha position: `absolute`, `fixed` ou `sticky`.

> **Contêiner Pai**: Certifique-se de que o elemento pai tenha algum atributo position não estático e dimensões definidas de alguma maneira.

>

> **Transformações Adicionais**: Certifique-se que, ao usar outros tipos de **configurações de transformação**, que elas não sobrescrevam os seus estilos!

---

Revisão #4
Criado 14 julho 2025 17:41:41 por Lyautey Maluf Neto
Atualizado 22 julho 2025 13:54:38 por Lyautey Maluf Neto