

# IMAGENS com CSS

5 PROPRIEDADES PARA MANIPULAÇÃO DE  
IMAGENS

GABRIELI MENDES NICOLODI

# CONTEÚDO

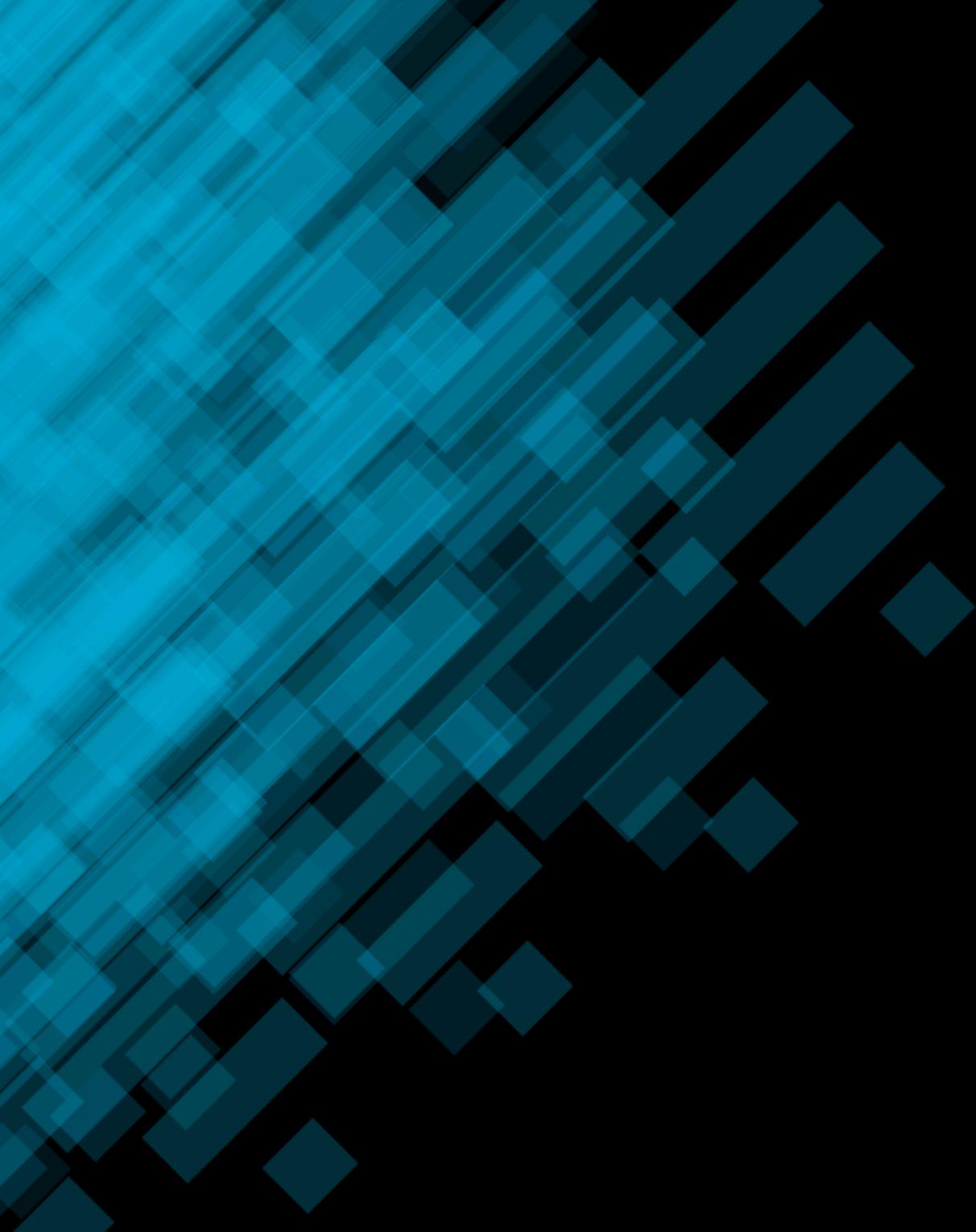
01 *MISTURANDO CAMADAS*

02 *RECORTE DE IMAGEM*

03 *APLICANDO FILTROS*

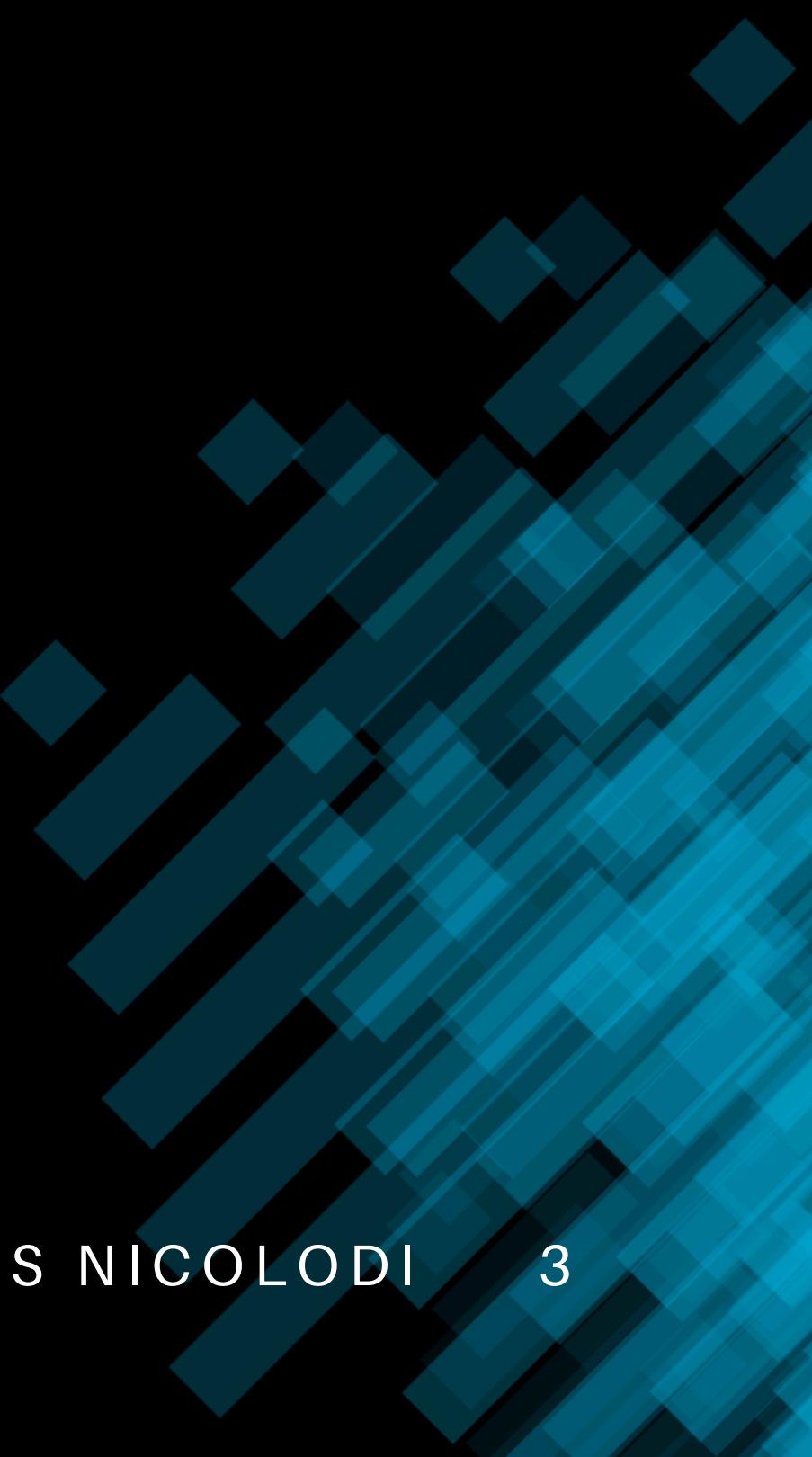
04 *APLICANDO MÁSCARAS*

05 *AJUSTE DE CONTEÚDO*



# CAPÍTULO 01 - MISTURANDO CAMADAS

*background-blend-mode*



# O QUE É BACKGROUND-BLEND-MODE?

A propriedade **background-blend-mode** define como as imagens de fundo e as cores de fundo se misturam entre si. Você pode criar efeitos visuais interessantes, mesclando uma imagem com uma cor sólida para obter diferentes nuances e tons.

## EXEMPLO DE APLICAÇÃO

```
/* Seleciona o elemento com a classe 'container' */
.container {
    /* Define a largura e altura do contêiner
    para visualizarmos o efeito */
    width: 300px;
    height: 200px;

    /* Define a cor de fundo do contêiner */
    background-color: blue;

    /* Define a imagem de fundo do contêiner */
    background-image: url('sua-imagem.jpg');

    /* Aplica o modo de mesclagem */
    background-blend-mode: multiply;

    /* Ajustando a imagem */
    background-size: cover;
    background-position: center;
}
```

# VALORES PARA BACKGROUND-BLEND-MODE

- *normal*: valor padrão;
- *multiply*: As cores são multiplicadas, resultando em uma cor mais escura.
- *screen*: As cores são invertidas, multiplicadas e novamente invertidas, resultando em uma cor mais clara.
- *overlay*: Combina *multiply* e *screen* dependendo da cor de fundo, escurecendo ou clareando as cores.
- *darken*: Mostra a cor mais escura entre as camadas.
- *lighten*: Mostra a cor mais clara entre as camadas.
- *color-dodge*: Aumenta o brilho da cor de fundo para refletir a cor da camada superior.
- *color-burn*: Escurece a cor de fundo para refletir a cor da camada superior.

## CAPÍTULO 02 - RECORTE DE IMAGEM

*clip-path*

# O QUE É CLIP-PATH?

A propriedade CSS **clip-path** permite recortar uma área específica de um elemento, como uma imagem, exibindo apenas a parte dentro da área definida. Isso é útil para criar formas complexas e efeitos visuais interessantes sem a necessidade de editar a imagem original.

## EXEMPLO DE APLICAÇÃO

```
.clip-image {  
    width: 300px;  
    height: 200px;  
    background-image: url('sua-imagem.jpg');  
    background-size: cover;  
    /* Define a forma de triângulo */  
    clip-path: polygon(50% 0%, 0% 100%, 100% 100%);  
}
```

# VALORES PARA CLIP-PATH

- *polygon()*: Define uma forma com base em uma lista de pontos (coordenadas). Cada ponto é um vértice da forma.
- *circle()*: Define uma área circular. O centro e o raio do círculo podem ser especificados.
- *ellipse()*: Define uma área elíptica. O raio horizontal e vertical, bem como o centro da elipse, podem ser especificados.
- *inset()*: Define uma área retangular com a possibilidade de arredondar os cantos. Os valores são os deslocamentos das bordas do contêiner.
- *path()*: Define uma forma utilizando comandos de caminho SVG (Scalable Vector Graphics). Isso permite formas muito complexas.
- *url()*: Refere-se a uma forma predefinida em um arquivo SVG externo. Você pode usar um URL para definir o recorte.

# CAPÍTULO 03 - APLICANDO FILTROS

*filter*

# O QUE É FILTER?

A propriedade CSS **filter** permite aplicar efeitos gráficos, como desfoque, brilho, contraste e mais, em elementos, incluindo imagens, textos e contêineres. Isso é útil para criar efeitos visuais sofisticados diretamente no CSS, sem a necessidade de editar a imagem original.

## EXEMPLO DE APLICAÇÃO

```
.filtered-image {  
    width: 300px;  
    height: 200px;  
    background-image: url('sua-imagem.jpg');  
    background-size: cover;  
    /* Aplica múltiplos filtros à imagem */  
    filter:  
        blur(5px)  
        brightness(150%)  
        contrast(200%)  
        grayscale(50%);  
}
```

# VALORES PARA FILTER

- *blur()*: Aplica um desfoque gaussiano.
- *brightness()*: Ajusta o brilho da imagem.
- *contrast()*: Ajusta o contraste da imagem.
- *drop-shadow()*: Aplica uma sombra projetada ao redor do elemento.
- *grayscale()*: Converte a imagem para tons de cinza.
- *hue-rotate()*: Ajusta o matiz da imagem.
- *invert()*: Inverte as cores da imagem.
- *opacity()*: Ajusta a opacidade da imagem.
- *saturate()*: Ajusta a saturação da imagem.
- *sepia()*: Converte a imagem para tons sépia.
- *url()*: Aplica um filtro gráfico de um recurso externo.

## CAPÍTULO 04 - APLICANDO MÁSCARAS

*mask*

# O QUE É MASK?

A propriedade CSS **mask** permite controlar a visibilidade de uma imagem ou de partes de um elemento usando uma máscara. A máscara define quais partes do elemento serão visíveis e quais serão ocultadas, criando efeitos visuais interessantes.

## EXEMPLO DE APLICAÇÃO

```
.masked-image {  
    width: 300px;  
    height: 200px;  
    background-image: url('sua-imagem.jpg');  
    background-size: cover;  
    mask-image: url('sua-mascara.png');  
    mask-size: cover;  
    mask-repeat: no-repeat;  
}
```

# VALORES PARA MASK

- *mask-image*: Define a imagem usada como máscara.
- *mask-mode*: Define o modo da máscara (alpha ou luminância).
- *mask-repeat*: Define como a imagem da máscara é repetida.
- *mask-position*: Define a posição da imagem da máscara.
- *mask-clip*: Define a área onde a máscara é aplicada.
- *mask-origin*: Define a origem da área da máscara.
- *mask-size*: Define o tamanho da imagem da máscara.
- *mask-composite*: Define o modo de composição da máscara.

## CAPÍTULO 05 - AJUSTE DE CONTEÚDO

*object-fit*

# O QUE É OBJECT-FIT?

A propriedade CSS **object-fit** define como o conteúdo de um elemento substituído, como uma imagem ou um vídeo, deve se ajustar às suas dimensões do contêiner.

Essa propriedade é particularmente útil para controlar a aparência de imagens ou vídeos quando o tamanho do contêiner difere do tamanho do conteúdo.

## EXEMPLO DE APLICAÇÃO

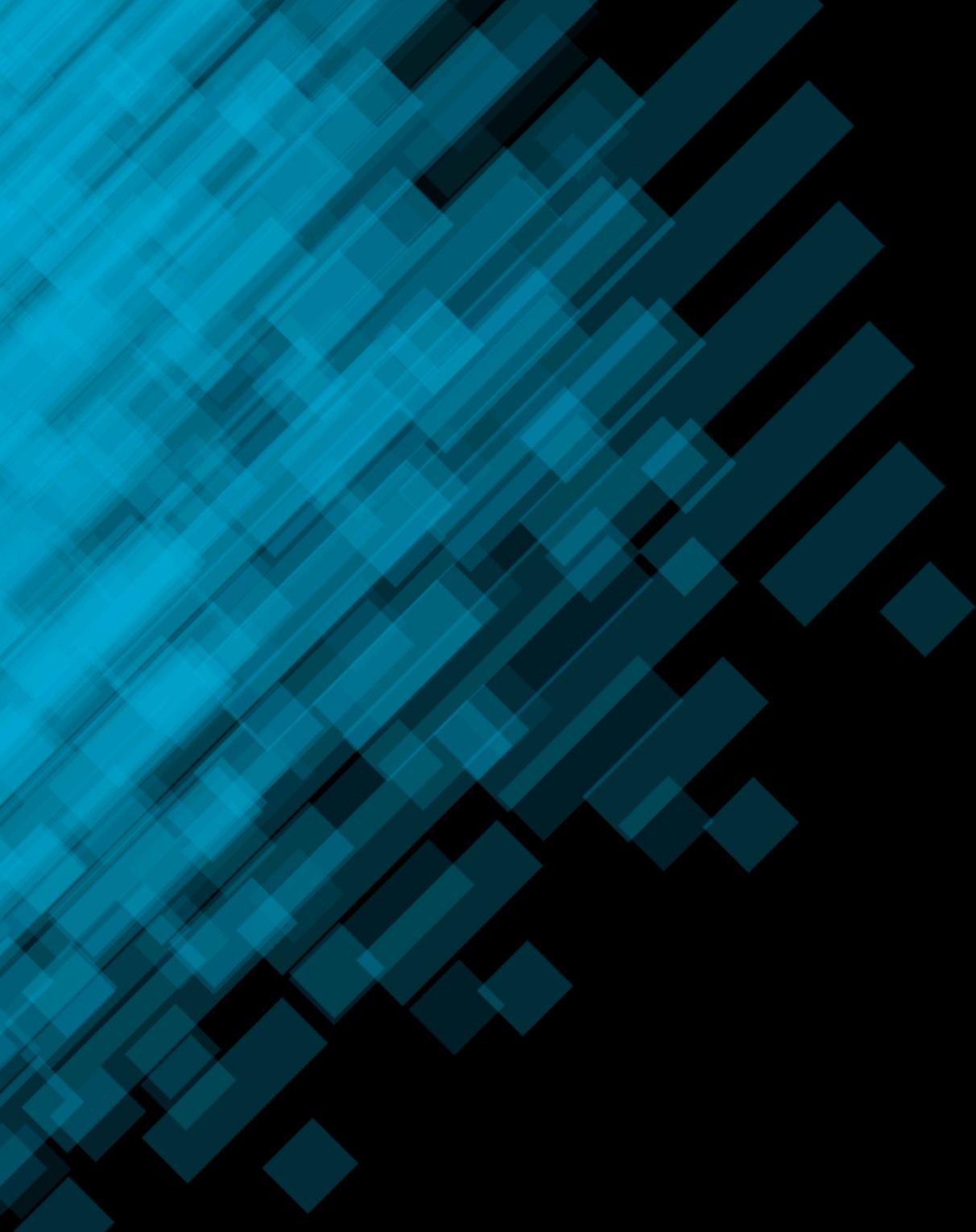
```
<div class="image-container">
  
</div>

.image-container {
  width: 300px;
  height: 200px;
  /* Esconde qualquer parte da imagem que saia do contêiner */
  overflow: hidden;
  /* Cor de fundo para visualização */
  background-color: #f0f0f0;
}

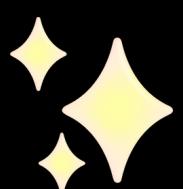
.fit-image {
  width: 100%;
  height: 100%;
  object-fit: cover;
}
```

# VALORES PARA OBJECT-FIT

- *fill*: O conteúdo preenche todo o contêiner, podendo distorcer se as proporções forem diferentes.
- *contain*: O conteúdo é redimensionado para caber dentro do contêiner, mantendo suas proporções. Pode haver espaço vazio nas laterais ou em cima/embaixo.
- *cover*: O conteúdo é redimensionado para cobrir completamente o contêiner, mantendo suas proporções. Parte do conteúdo pode ser cortada.
- *none*: O conteúdo mantém seu tamanho original, independentemente do tamanho do contêiner.
- *scale-down*: Comporta-se como *none* ou *contain*, o que quer que faça o conteúdo menor.



## CAPÍTULO 06 - AGRADECIMENTO



# OBRIGADA POR LER ATÉ AQUI!

Esse e-book foi gerado por IA e diagramado por um humano.

O passo a passo e ferramentas utilizadas se encontram no meu Github.

Esse e-book foi gerado como parte da trilha Fundamentos de IA para Devs do Santander Bootcamp 2024 na plataforma DIO.



<https://github.com/GabrieliMendesNicolodi/create-ebook-with-ai>

## CONECTE-SE COM A AUTORA



✉ gabiprogramadoraweb@gmail.com  
LinkedIn [linkedin.com/in/gabiprogramadoraweb/](https://linkedin.com/in/gabiprogramadoraweb/)