

Na aula passada criamos a interface gráfica do nosso app filtro de imagem, customizando nossa cena principal com um UIImageView (mostrando uma imagem nele), e um botão. Bom, vamos agora ver como construir o código do app e dar uma funcionalidade a essa botão.

#01: Vá na StoryBoard, selecione o UIImageView e no “attributes inspector” deixe vazio o campo Image para colocarmos a imagem no UIImageView direto pelo código.

#02: No canto superior direito clique no ícone onde mostram 2 círculos chamado “Show the Assistant Editor”. Isso irá dividir sua tela em 2, uma para a Storyboard e outra para o código do arquivo da cena que você tem selecionada, no caso, a cena ViewController.

#03: Antes de mais nada, precisamos conectar o elemento de interface, o UIImageView, ao nosso código, para isso selecione o UIImageView e com control pressionado arraste-o com o trackpad para qualquer lugar entre a classe e o método viewDidLoad(). Ao soltar dê o nome de imageView à variável e Done. O @IBOutlet indica que a variável está conectada a um elemento na interface. Daqui a pouco iremos usar o UIImageView no nosso código.

#04: Crie 3 variáveis abaixo do imageView:

A primeira armazenará a imagem original

```
var imagemSemFiltro = UIImage(named: "bebe.jpg")
```

A segunda para mais tarde armazenar a imagem após esta ter passado pelo filtro. Usamos o ponto de ! após o tipo para dizer que ela é uma optional até possuir um valor, depois disso não é preciso usar o ponto de ! para desempacotar seu valor.

```
var imagemComFiltro: UIImage!
```

E uma terceira para armazenar o filtro futuramente

```
var filtro1: CIColor!
```

#05: Vamos utilizar apenas o método viewDidLoad() que é responsável por carregar os dados iniciais do App antes da cena aparecer para o usuário. Tudo abaixo deve ser acrescentado nesse método.

Primeiro vamos apresentar a imagem ao usuário. Para isso vamos usar o UIImageView que conectamos ao código.

```
imageView.image = imagemSemFiltro
```

Agora vamos preparar a imagem para passar pelo filtro. Lembrando que para passar por filtros, as imagens precisam ser do tipo CImage e não UIImage.

```
let imagemParaFiltro = CImage(image: imagemSemFiltro)
```

Logo em seguida, vamos criar o filtro

```
filtro = CIFilter(name: "CISepiaTone", withInputParameters: ["inputImage":imagemParaFiltro,  
"inputIntensity": 1.0])
```

E por último guardar o valor da imagem que passou pelo filtro. Como imagemComFiltro é do tipo UIImage, precisamos converter a imagem de saída do filtro que é uma CImage para UIImage.

```
imagemComFiltro = UIImage(CImage: filtro.outputImage)
```

#06: Já carregamos a imagem na interface para o usuário e sem ele saber já geramos a imagem com filtro. Agora precisamos que ao clicar no botão Aplicar Filtro seja mostrada a imagem com o filtro. Para isso vamos fazer o mesmo procedimento que fizemos com o UIImageView, mas desta vez vamos selecionar o botão Aplicar Filtro e com control + trackpad arrastar até o código, coloque após o fecha chaves do viewDidLoad(). Ao soltar, escolha a opção Action em Connection e dê o nome de aplicarFiltro, em Type escolha UIBarButtonItem (que é o tipo do botão) e Done. Nisso foi gerado para nós um método responsável por fazer algo quando a pessoa clicar no botão. O @IBAction indica que o método deve ser executado quando o elemento de interface ao qual ele está ligado seja pressionado pelo usuário.

Dentro dele colocaremos a seguinte linha de código que carregará a imagem com o filtro no UIImageView quando o usuário clica no botão.

```
imageView.image = imagemComFiltro
```

#07: Agora é só mandar o App rodar e ver o resultado.