

#01: Crie mais um botão chamado Limpar que retire o filtro aplicado na imagem mostrando a imagem antiga.

Resposta:

Criar o bloco abaixo da mesma forma como foi conectado o IBAction do passo a passo.

```
@IBAction func limpar(sender: UIBarButtonItem) {  
  
    imageView.image = imagemSemFiltro  
  
}
```

#02: Renomeie o botão de Aplicar Filtro na interface para Filtro1 e crie um segundo botão chamado Filtro2, e dependendo de qual botão o usuário clicar seja mostrada uma imagem com um determinado filtro. Renomeie também a variável filtro para `filtro1` e crie um outra chamada `filtro2` para armazenar o novo filtro.

O novo filtro pode ser o mesmo utilizado no desafio da aula 03, o `CIHueAdjust` com os parâmetros `inputImage` e `inputAngle` especificados na documentação: [\[link\]](#).

*Dica1: `imagemRenderizada` e `imagemComFiltro` não poderão mais ser usados no `viewDidLoad()`

*Dica2: se você for renomear o método `aplicarFiltro` para `aplicarFiltro1`, lembrar de usar `control + click` em cima do botão Filtro1 e excluir a conexão com o método, então ir até a bolinha do lado do método e pressionar `option` e com o trackpad arrastar e soltar em cima do botão Filtro1 para a conexão ser reestabelecida.

Resposta #02:

```
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var imageView: UIImageView!
14
15     var imagemSemFiltro = UIImage(named: "bebe.jpg")
16     var imagemComFiltro: UIImage!
17
18     var filtro1: CIFilter!
19     var filtro2: CIFilter!
20
21     let context = CIContext(options: nil)
22
23     override func viewDidLoad() {
24         super.viewDidLoad()
25
26         imageView.image = imagemSemFiltro
27
28         let imagemParaFiltro = CIImage(image: imagemSemFiltro)
29
30         filtro1 = CIFilter(name: "CISepiaTone", withInputParameters: ["inputImage": imagemParaFiltro, "inputIntensity": 1.0])
31         filtro2 = CIFilter(name: "CIHueAdjust", withInputParameters: ["inputImage": imagemParaFiltro, "inputAngle": 5.0])
32     }
33
34     @IBAction func aplicarFiltro1(sender: UIBarButtonItem) {
35
36         let imagemRenderizada = context.createCGImage(filtro1.outputImage, fromRect: filtro1.outputImage.extent())
37
38         imagemComFiltro = UIImage(CGImage: imagemRenderizada)
39         imageView.image = imagemComFiltro
40     }
41
42
43
44
45     @IBAction func limpar(sender: UIBarButtonItem) {
46
47         imageView.image = imagemSemFiltro
48     }
49
50
51     @IBAction func aplicarFiltro2(sender: UIBarButtonItem) {
52
53         let imagemRenderizada = context.createCGImage(filtro2.outputImage, fromRect: filtro2.outputImage.extent())
54
55         imagemComFiltro = UIImage(CGImage: imagemRenderizada)
56         imageView.image = imagemComFiltro
57     }
58 }
```

#03: Ao invés de usar 2 constantes, uma para um filtro e outra para outro, crie uma constante chamada `filtros` para armazenar os 2 filtros. Escolha o tipo de coleção que melhor lhe for útil.

*Dica: Se optar por uma `Array`, lembrar de inicializá-la com `[]` para poder usar seu método `append()` para adicionar os filtros.

Resposta #03.1 - com Array:

```
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var imageView: UIImageView!
14
15     var imagemSemFiltro = UIImage(named: "bebe.jpg")
16
17     var imagemComFiltro: UIImage!
18
19     var filtros: [CIFilter] = []
20
21     override func viewDidLoad() {
22         super.viewDidLoad()
23         // Do any additional setup after loading the view, typically from a nib.
24
25         imageView.image = imagemSemFiltro
26
27         let imagemParaFiltro = UIImage(image: imagemSemFiltro)
28
29         filtros.append(CIFilter(name: "CISepiaTone", withInputParameters: ["inputImage": imagemParaFiltro, "inputIntensity": 1.0]))
30
31         filtros.append(CIFilter(name: "CIHueAdjust", withInputParameters: ["inputImage": imagemParaFiltro, "inputAngle": 5.0]))
32
33     }
34
35     @IBAction func aplicarFiltro1(sender: UIBarButtonItem) {
36
37         imagemComFiltro = UIImage(CIImage: filtros[0].outputImage)
38
39         imageView.image = imagemComFiltro
40
41     }
42
43
44     @IBAction func limpar(sender: UIBarButtonItem) {
45
46         imageView.image = imagemSemFiltro
47
48     }
49
50
51     @IBAction func aplicarFiltro2(sender: UIBarButtonItem) {
52
53         imagemComFiltro = UIImage(CIImage: filtros[1].outputImage)
54
55         imageView.image = imagemComFiltro
56
57     }
58 }
```

Resposta #03.2 - com Dicionário:

```
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var imageView: UIImageView!
14
15     var imagemSemFiltro = UIImage(named: "bebe.jpg")
16
17     var imagemComFiltro: UIImage!
18
19     var filtros: [String: CIFilter] = [:]
20
21     override func viewDidLoad() {
22         super.viewDidLoad()
23         // Do any additional setup after loading the view, typically from a nib.
24
25         imageView.image = imagemSemFiltro
26
27         let imagemParaFiltro = CIImage(image: imagemSemFiltro)
28
29         filtros["Sepia"] = CIFilter(name: "CISepiaTone", withInputParameters: ["inputImage": imagemParaFiltro, "inputIntensity": 1.0])
30
31         filtros["Hue"] = CIFilter(name: "CIHueAdjust", withInputParameters: ["inputImage": imagemParaFiltro, "inputAngle": 5.0])
32     }
33
34
35     @IBAction func aplicarFiltro1(sender: UIBarButtonItem) {
36
37         //Lembrando que o valor retornado por uma chave no dicionário é uma optional por isso temos que desempacotá-lo!
38
39         imagemComFiltro = UIImage(CIImage: filtros["Sepia"]!.outputImage)
40
41         imageView.image = imagemComFiltro
42     }
43
44
45
46     @IBAction func limpar(sender: UIBarButtonItem) {
47
48         imageView.image = imagemSemFiltro
49     }
50
51
52
53     @IBAction func aplicarFiltro2(sender: UIBarButtonItem) {
54
55         imagemComFiltro = UIImage(CIImage: filtros["Hue"]!.outputImage)
56
57         imageView.image = imagemComFiltro
58     }
59 }
```