

Passando método como parâmetro

Abaixo, temos uma representação do método `mostrarDetalhes`. Nele, criamos um `UIAlertAction` para remover a refeição selecionada.

```
@objc func mostrarDetalhes(_ gesture: UILongPressGestureRecognizer) {  
    if gesture.state == .began {  
        let botaoRemover = UIAlertAction(title: "remover", style: .destructive, handler: {  
  
            // Resto da implementação do método  
  
        })  
    }  
}
```

Para poder remover a refeição selecionada, precisamos dizer à action como ela deve fazer isso, portanto, no parâmetro `handle` do construtor da classe `UIAlertAction`, vamos substituir o `nil` por um método. Mas, onde devemos criar esse método?

Selecione uma alternativa

A

Criamos o método dentro da classe.

```
func removerSelecionado(action: UIAlertAction) {  
    refeicoes.remove(at: row)  
    tableView.reloadData()  
}  
  
@objc func mostrarDetalhes(_ gesture: UILongPressGestureRecognizer) {  
    if gesture.state == .began {  
        let botaoRemover = UIAlertAction(title: "remover", style: .destructive, handl  
  
        // Resto da implementação do método  
  
    }  
}
```

B

Criamos o método dentro do `mostrarDetalhes`.

```
@objc func mostrarDetalhes(_ gesture: UILongPressGestureRecognizer) {  
    if gesture.state == .began {  
        func removerSelecionado(action: UIAlertAction) {  
            refeicoes.remove(at: row)  
            tableView.reloadData()  
        }  
        let botaoRemover = UIAlertAction(title: "remover", style: .destructive, handl  
        // Resto da implementação do método  
  
    }  
}
```

C

Criamos o método dentro do próprio parâmetro.

```
@objc func mostrarDetalhes(_ gesture: UILongPressGestureRecognizer) {  
    if gesture.state == .began {  
        let botaoRemover = UIAlertAction(title: "remover", style: .destructive, handler: {  
            self.refeicoes.remove(at: row)  
            self.tableView.reloadData()  
        })  
  
        // Resto da implementação do método  
    }  
}
```

