



Gestão de memória e descobertas de Leaks



I'M A LOVER ♥ REVOLUCIONAR O UNIVERSO DA ALIMENTAÇÃO



É M A LOVER ♥ REVOLUCIONAR O UNIVERSO DA ALIMENTAÇÃO

Renata Faria

Desenvolvedora iOS @ iFood

Organizadora Cocoaheads - Recife

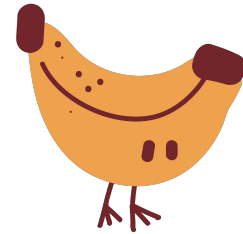
Apple Developer Academy Alumni

dúvidas?
slido.com
#A451



Agenda

1. O problema
2. Gerência de memória e ARC
3. Ciclo de referências
4. Tipos de ligação
5. Vida Real
6. Ferramentas





`let tdcSP = Event()`





let tdcSP = Event()



Alocação

é reservado um
espaço dentro da
memória para salvar
aquele elemento



let tdcSP = Event()



Alocação

é reservado um
espaço dentro da
memória para salvar
aquele elemento



Inicialização

antes do elemento ser
utilizado, ele será
inicializado e assim salvo
no espaço alocado



let tdcSP = **Event**()



let other = **Class**()





let tdcSP = **Event**()





let tdcSP = **Event**()



let other = **Class**()







`let tdcRecife = Event()`





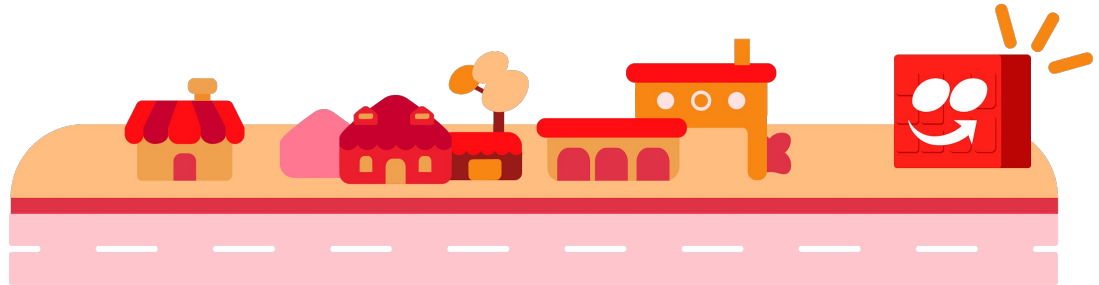
```
let tdcRecife = Event()
```





Gerência de memória

**"Jogar fora" o que
não está sendo mais
usado**



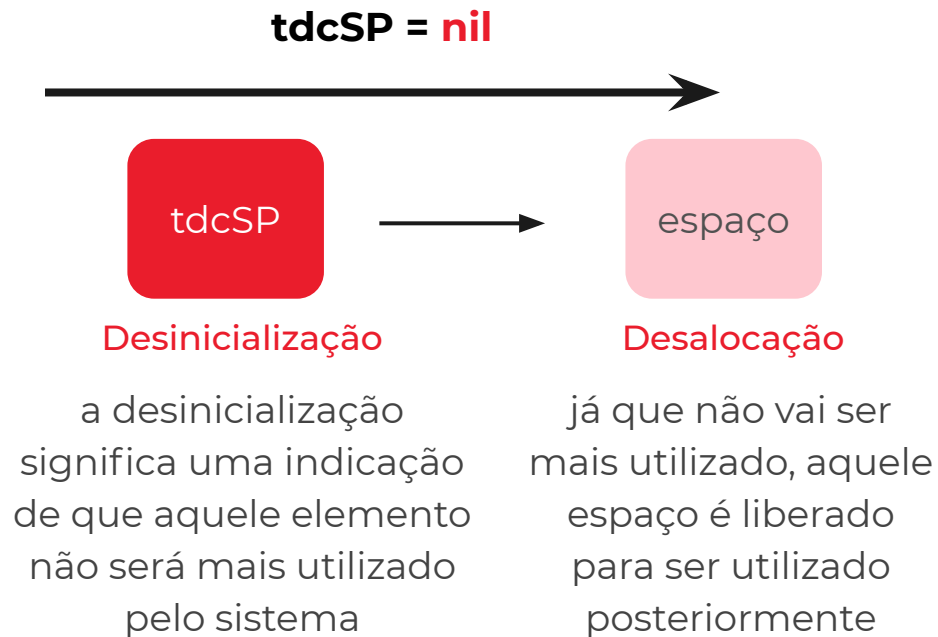


var tdcSP = Event()



No domingo a tdc sp já vai ter terminado, então será importante esvaziar esse espaço para ser utilizado por outras conferências







**O desalocamento é
automático?**



I'M A LOVER ❤️ REVOLUCIONAR O UNIVERSO DA ALIMENTAÇÃO

ARC

Automatic
Reference
Counting

**Contar quantas
"ligações" tem
naquele espaço na
memória**



Ação

var tdcSP = **Event**()



ARC

1

Ação

ARC

var tdcSP = **Event**()



1

tdcSP = **nil**



0

Ação

var tdcSP = **Event**()



ARC

1

Se tiver zero, então desaloco!

tdcSP = **nil**



0

Desalocado

Por outro ângulo..

```
var tdcSP = Event()
```



ligações: 1

Espaço
preenchido

espaço

Por outro ângulo..

tdcSP = nil

ligações: 0

espaço

espaço

Mas a TDC de Recife é igual, vamos reaproveitar a TDC SP..

Mas a TDC de Recife é igual, vamos reaproveitar a TDC SP..

```
var tdcRec = tdcSP
```

Mas a TDC de Recife é igual, vamos reaproveitar a TDC SP..

```
var tdcRec = tdcSP
```

lembre-se: classes são
reference type

Por outro ângulo..

```
var tdcSP = Event()
```



ligações: 1

Espaço
preenchido

espaço

Por outro ângulo..

```
var tdcSP = Event()
```

```
var tdcRec = tdcSP
```



ligações: 2

Espaço
preenchido

espaço

Por outro ângulo..

tdcSP = nil

var tdcRec = tdcSP



ligações: 1

Espaço
preenchido

espaço

Por outro ângulo..

tdcSP = nil

tdcRec = nil

ligações: 0

espaço

espaço

Ação

var tdcSP = **Event**()



ARC - tdcSP

1

Ação

ARC - tdcSP

var tdcSP = **Event**()



1

var tdcRec = tdcSP



2

Ação

var tdcSP = **Event**()



var tdcRec = tdcSP



tdcSP = **nil**



ARC - tdcSP

1

2

1

não será
Desalocado

Ação

var tdcSP = **Event()**



var tdcRec = tdcSP



tdcSP = **nil**



tdcRec = **nil**



ARC - tdcSP

1

2

1

0

não será
Desalocado

Desaloca tanto a
tdcRec quando
tdcSP



**E por que
saber disso
importa?**

I'M A LOVER ❤️ REVOLUCIONAR O UNIVERSO DA ALIMENTAÇÃO

Leak

Um objeto que não está sendo utilizado mas ocupa espaço na memória por não ter sido desalocado

Nova classe!

```
var you = Attendee()
```



Nova classe!

```
var you = Attendee()
```

Faz sentido um participante se não tem um evento???



Nova classe!

`var you = Attendee()`

Faz sentido um participante se não tem um evento???

`var you = Attendee(event: tdcSP)`



E as ligações?

```
var tdcSP = Event()
```



ligações: 1

tdcSP

ligações: 0

espaço

E as ligações?

```
var tdcSP = Event()
```



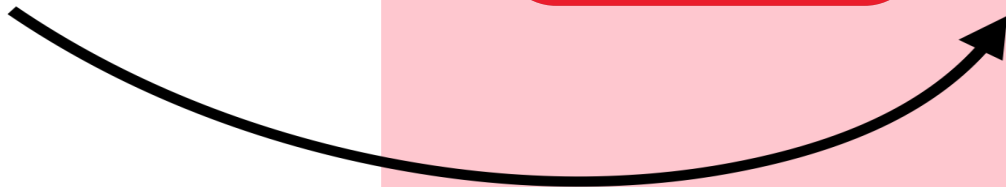
ligações: 1

tdcSP

ligações: 1

you

```
var you = Attendee(event: tdcSP)
```



E as ligações?

```
var tdcSP = Event()
```

```
var you = Attendee(event: tdcSP)
```

ligações: 2

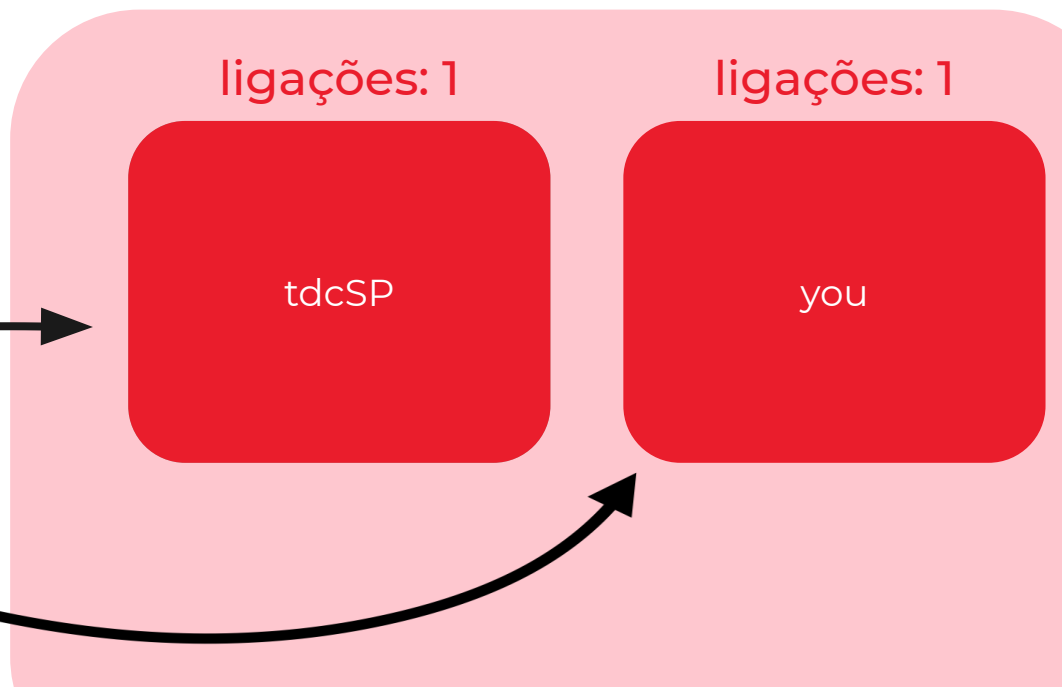
tdcSP

ligações: 1

you

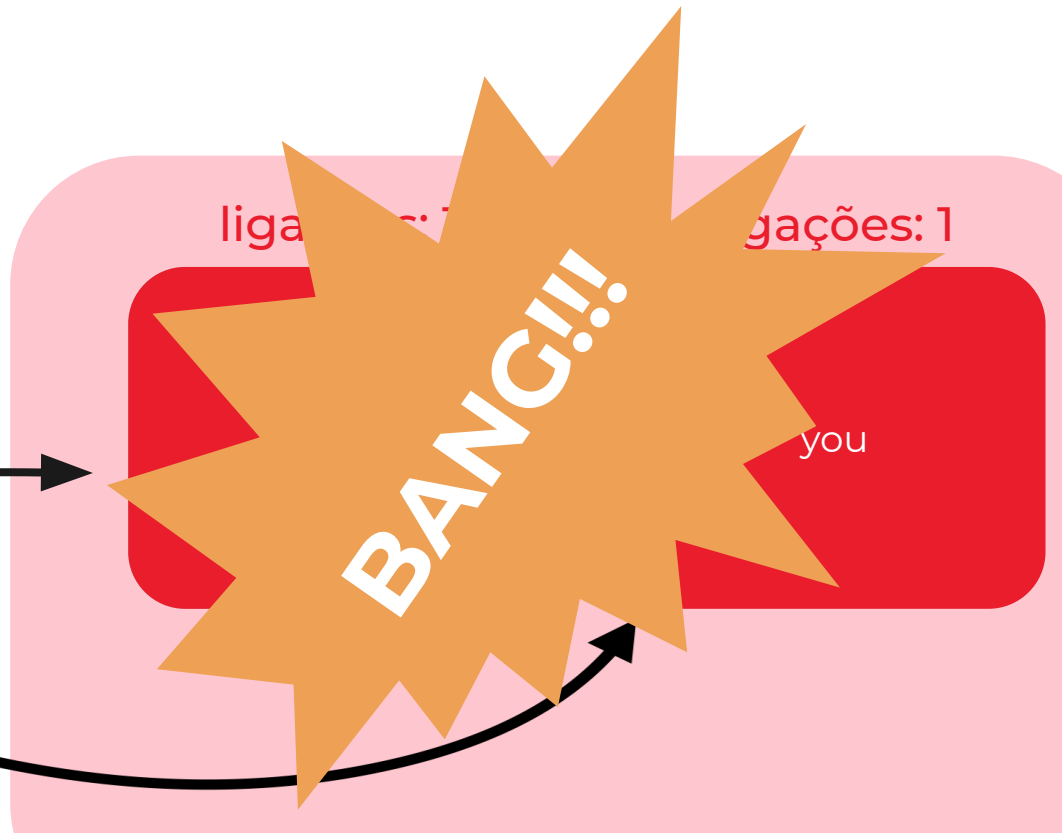
E as ligações?

```
tdcSP = nil  
  
var you = Attendee(event: tdcSP)
```



E as ligações?

```
tdcSP = nil  
  
var you = Attendee(event: tdcSP)
```



**Ligações entre
podem prender
objetos na memória!**



Retain Cycle

Uma referência aponta para a outra e vice-versa, impedindo o ARC de tirá-las da memória

```
var you = Attendee(event: tdcSP)
```



```
var you = Attendee(event: tdcSP)
```

Tem um espacinho só para você 🥰

```
tdcSP.attendee = you
```

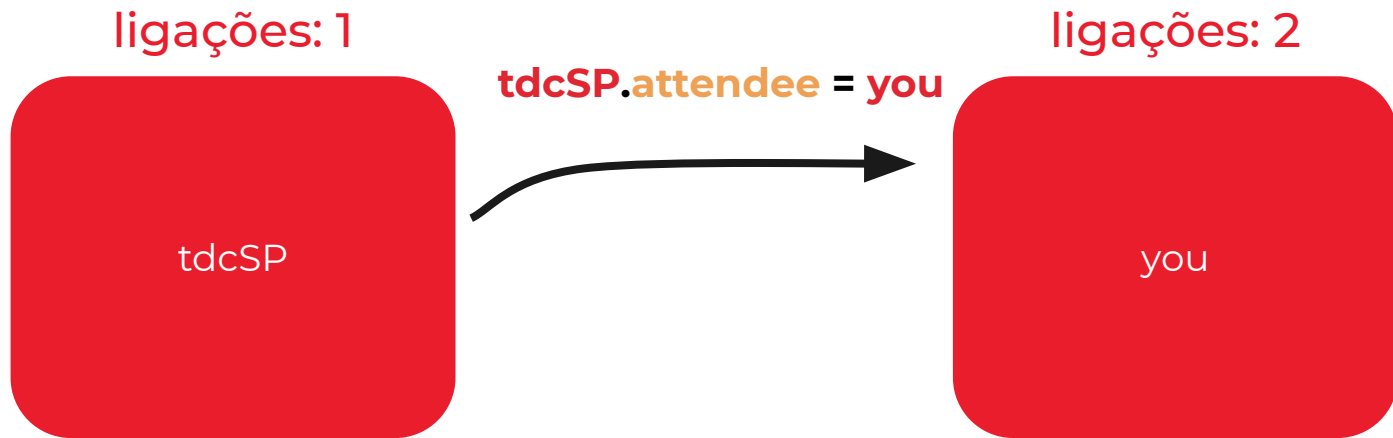


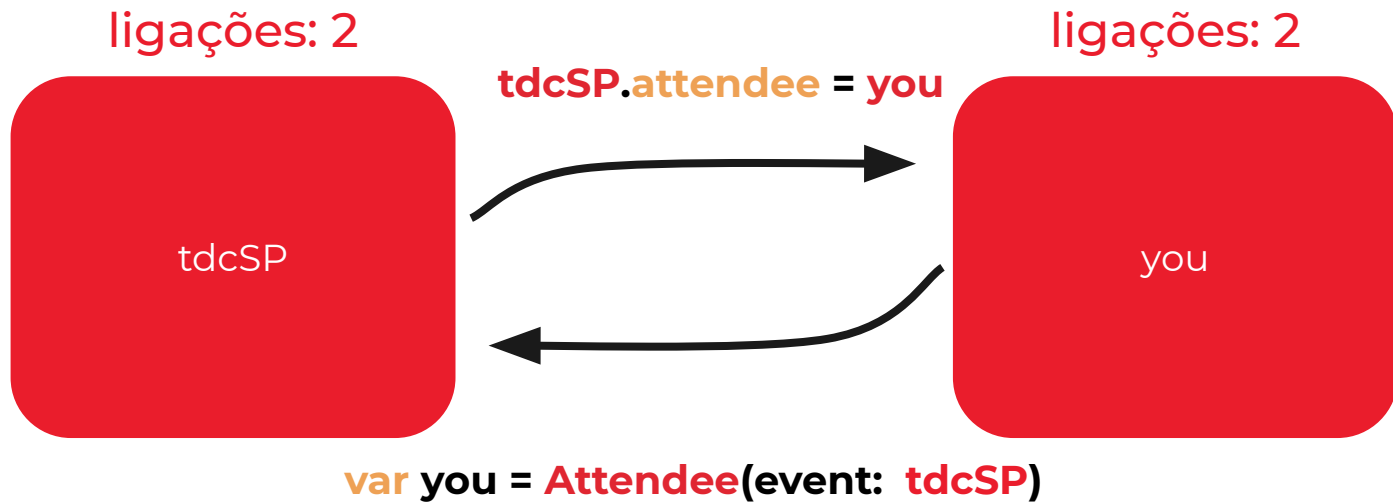
ligações: 1

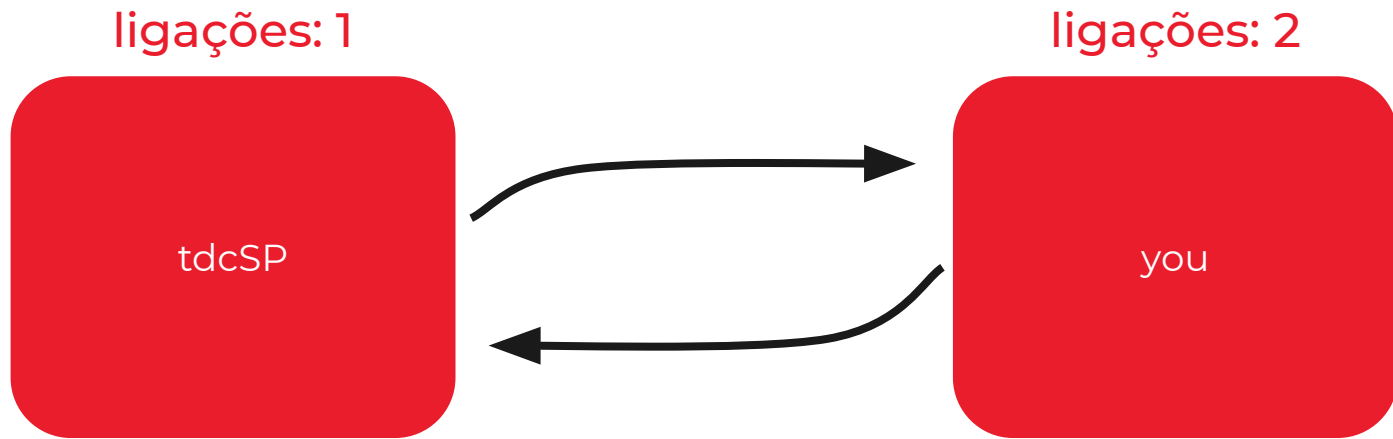
tdcSP

ligações: 1

you

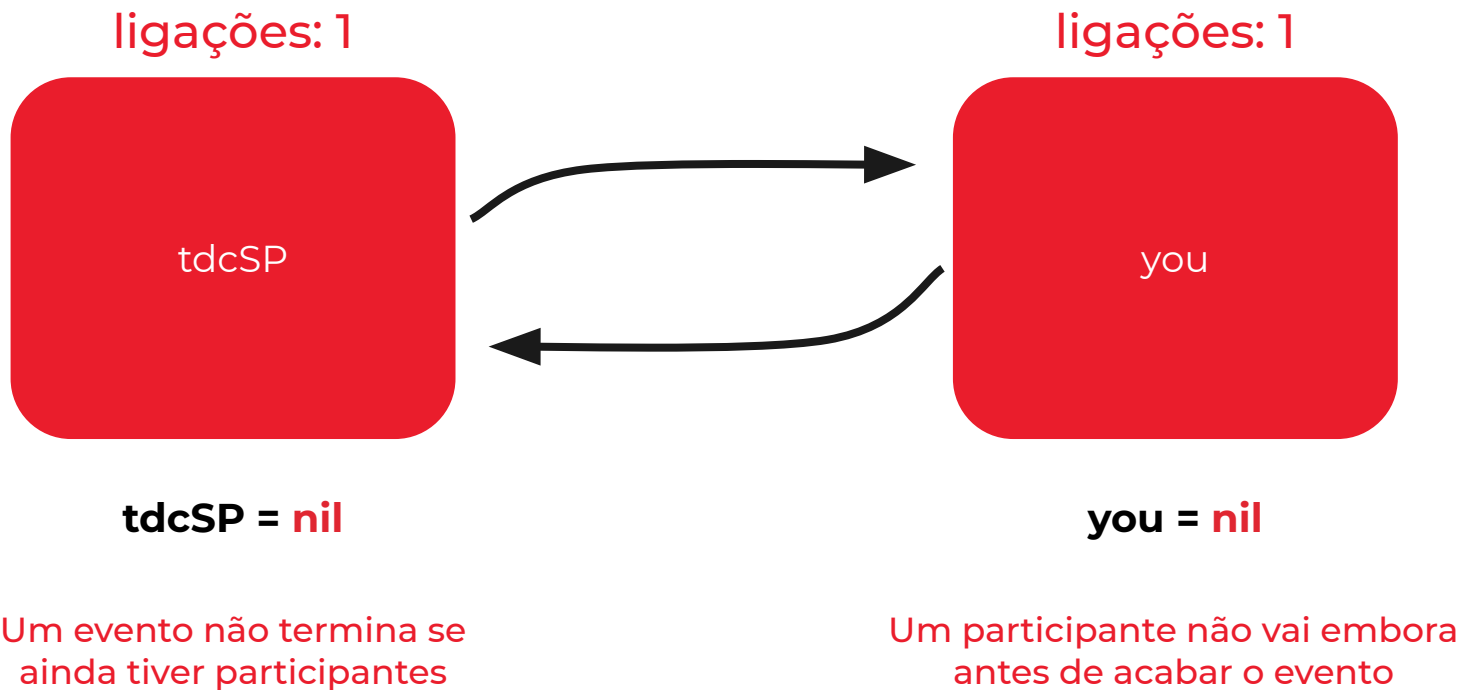






tdcSP = nil

Um evento não termina se
ainda tiver participantes

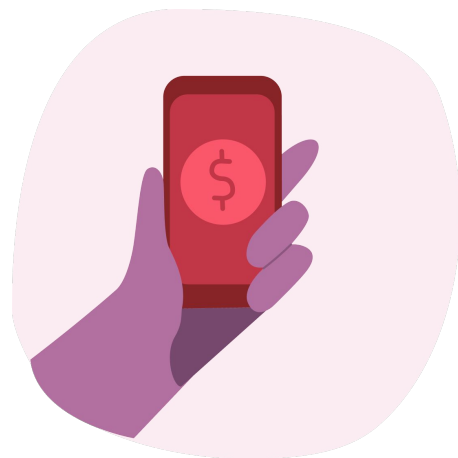


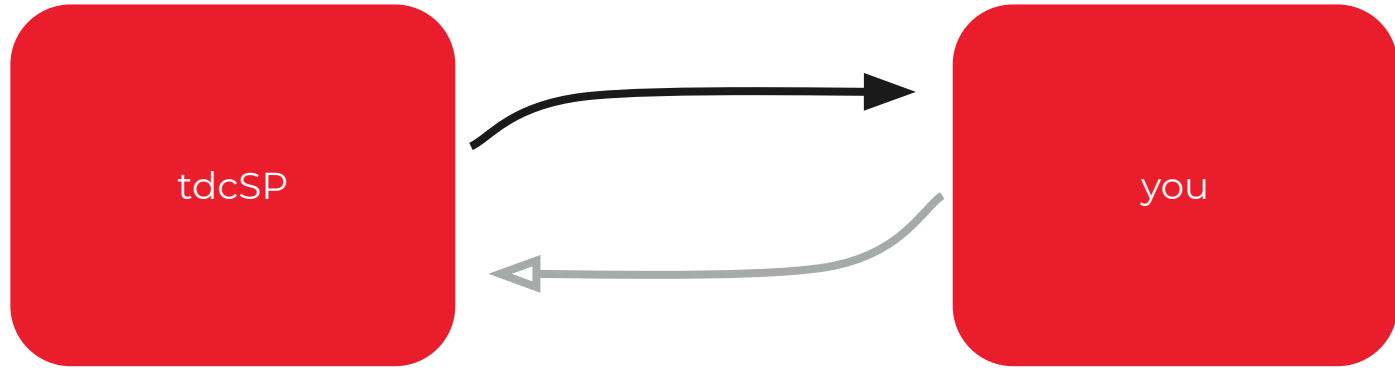


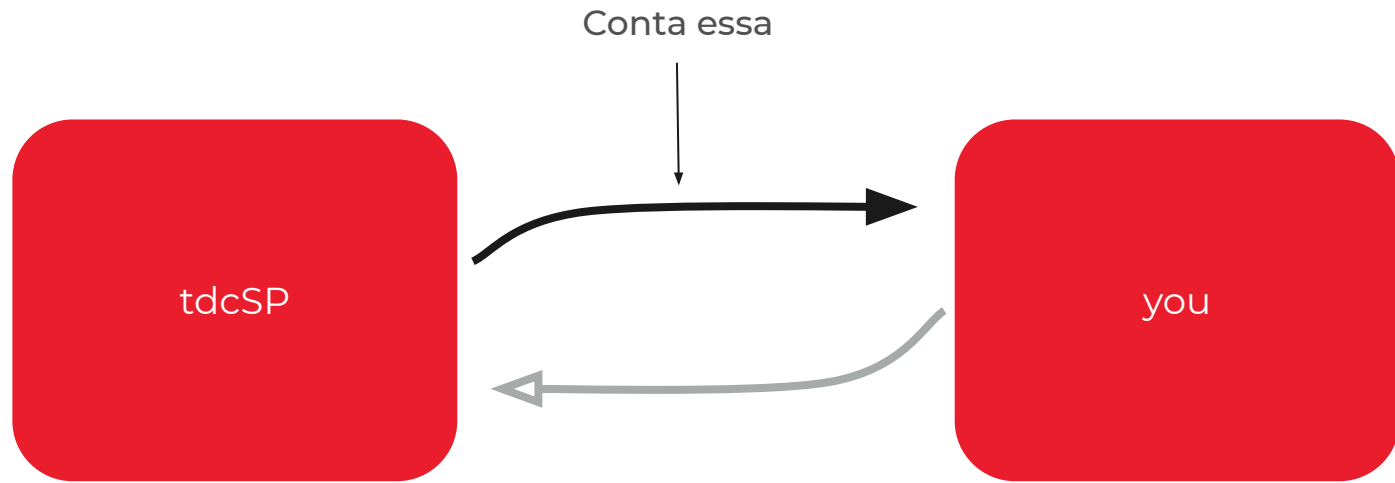
**E como
resolver
isso?**

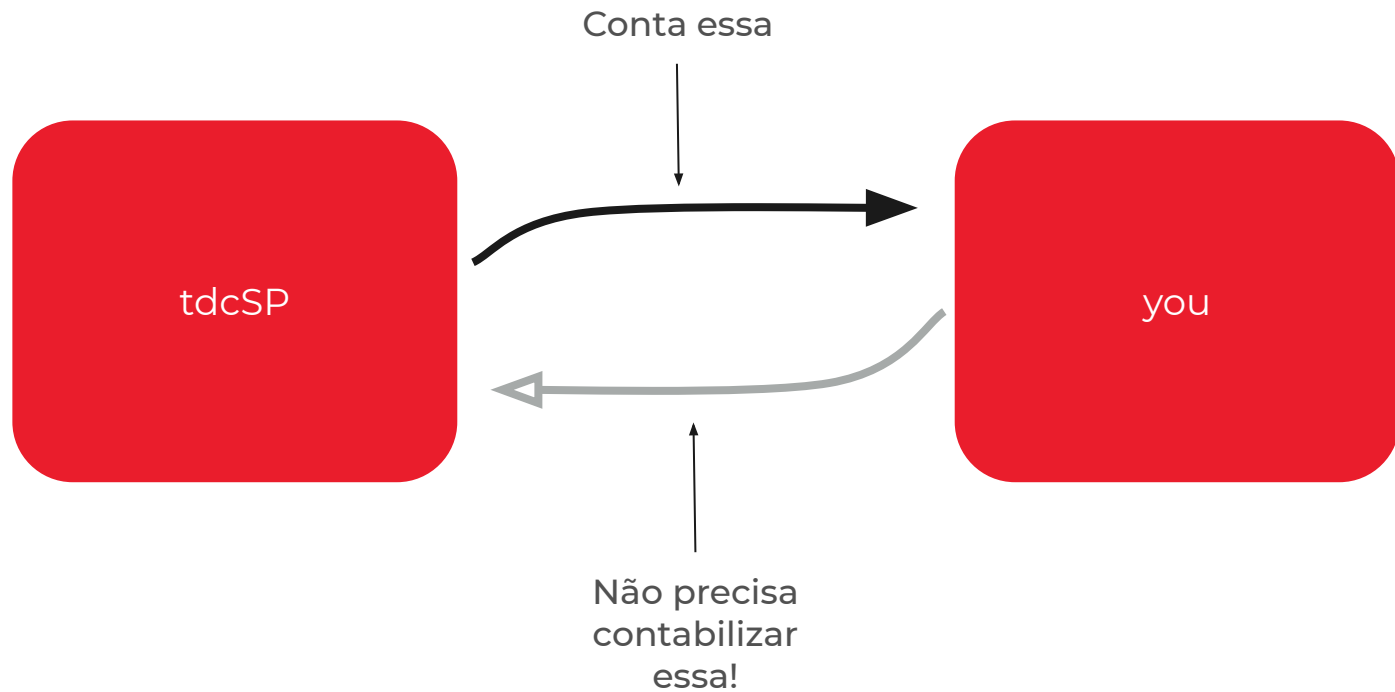
I'M A LOVER ❤️ REVOLUCIONAR O UNIVERSO DA ALIMENTAÇÃO

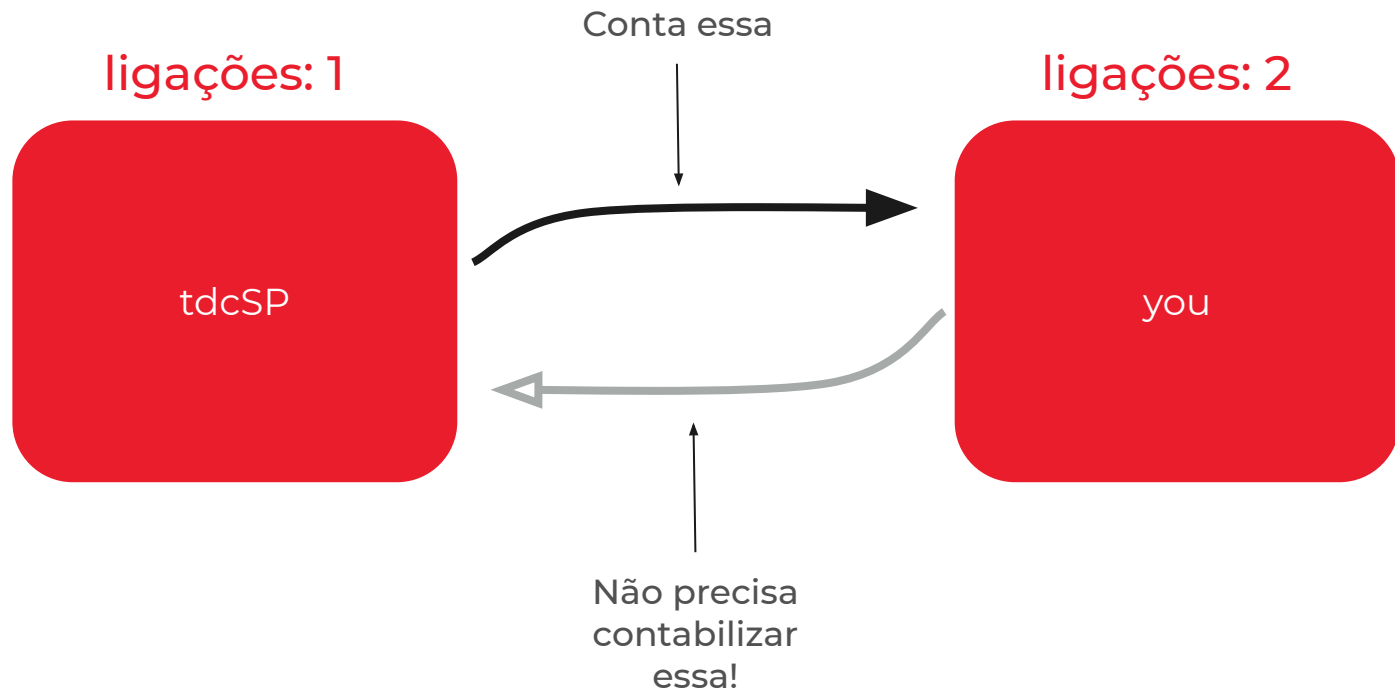
**Avisar para o ARC
que aquela ligação
não precisa ser
contabilizada**

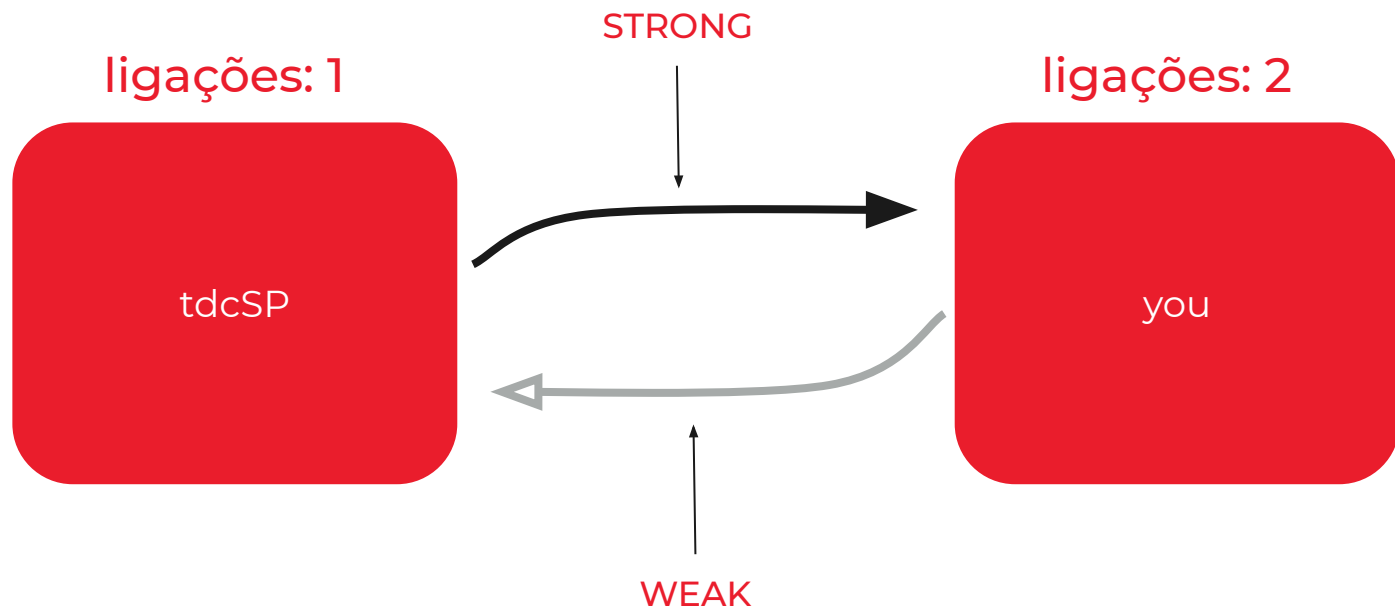












ligações: 0



tdcSP = nil

ligações: 1



ligações: 0



tdcSP = nil

ligações: 1



Essa ligação
ainda existe!!

ligações: 0



tdcSP = nil

ligações: 1



you.event.name



ligações: 0



espaço

tdcSP = nil



ligações: 0

nome

Efeito colateral

**Uma ligação fraca
torna o objeto
opcional**



ligações: 0



tdcSP = nil

ligações: 1



you.event?.name





**Mas não
quero ficar
usando ?**

Opção safe:

```
guard let strongTdc = tdc else { return }
```

Opção safe:

```
guard let strongTdc = tdc else { return }
```

Opção não-safe: (se você souber usar, é mais eficiente)

Utilizar a ligação do tipo unowned

Opção safe:

```
guard let strongTdc = tdc else { return }
```

Opção não-safe: (se você souber usar, é mais eficiente)

Utilizar a ligação do tipo unowned



Ligações "sem dono" são ligações não contabilizadas pelo ARC mas que não são opcionais...

Opção safe:

```
guard let strongTdc = tdc else { return }
```

Opção não-safe: (se você souber usar, é mais eficiente)

Utilizar a ligação do tipo unowned



Ligações "sem dono" são ligações não contabilizadas pelo ARC mas que não são opcionais...

Force-unwrap em ligações weak



**Como esse
problema
aparece na
vida real??**

I'M A LOVER ❤️ REVOLUCIONAR O UNIVERSO DA ALIMENTAÇÃO

**Em vários! Mas a
grande maioria
aparecem em
protocolos e
closures**

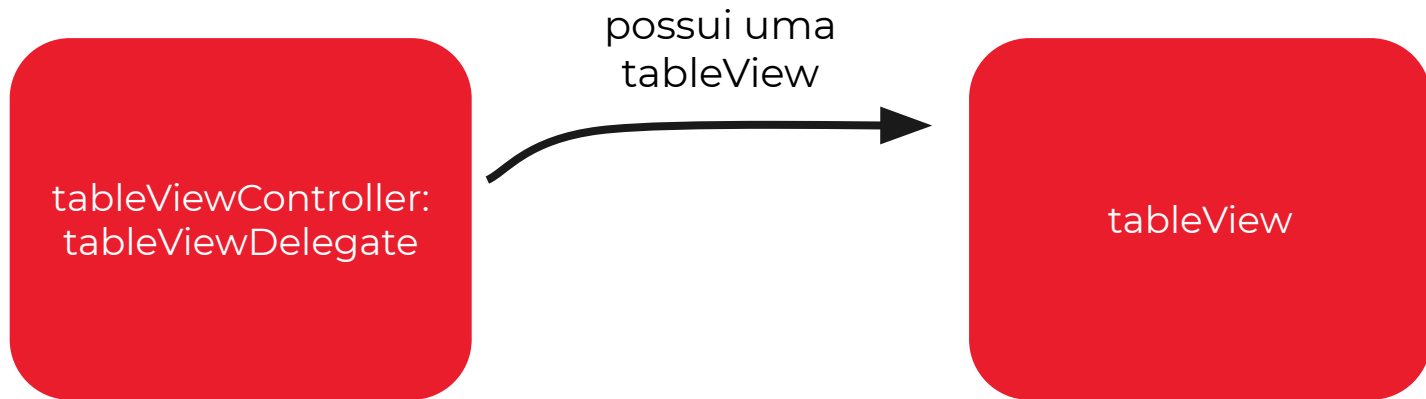


1. Protocolos

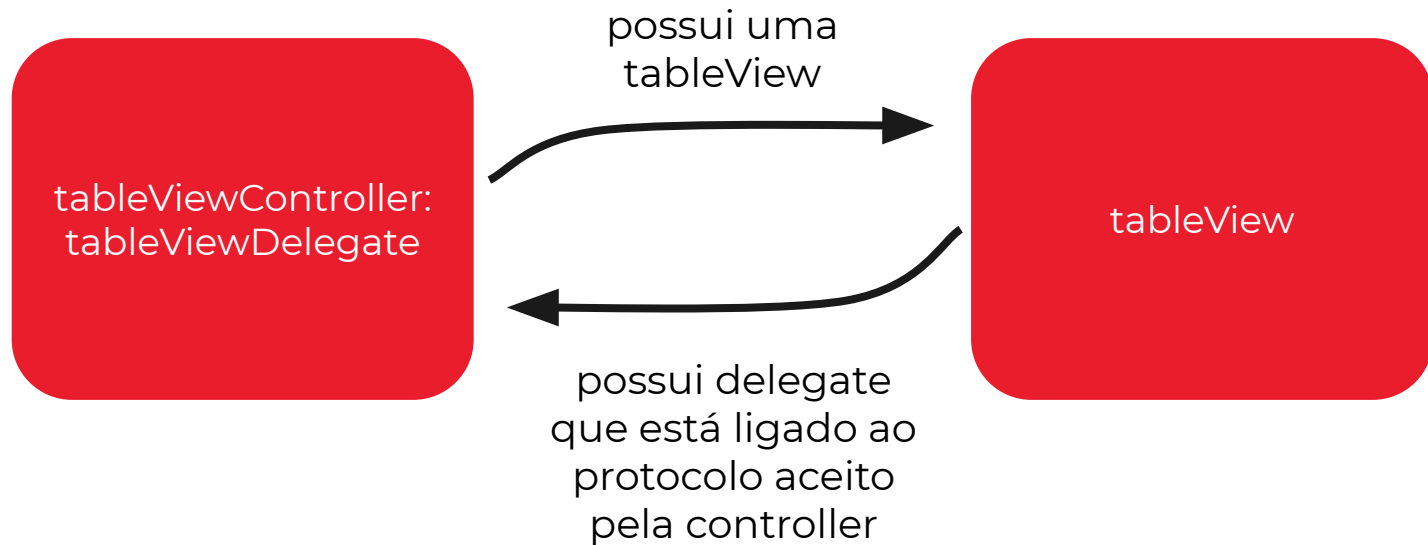
tableViewController:
tableViewDelegate

tableView

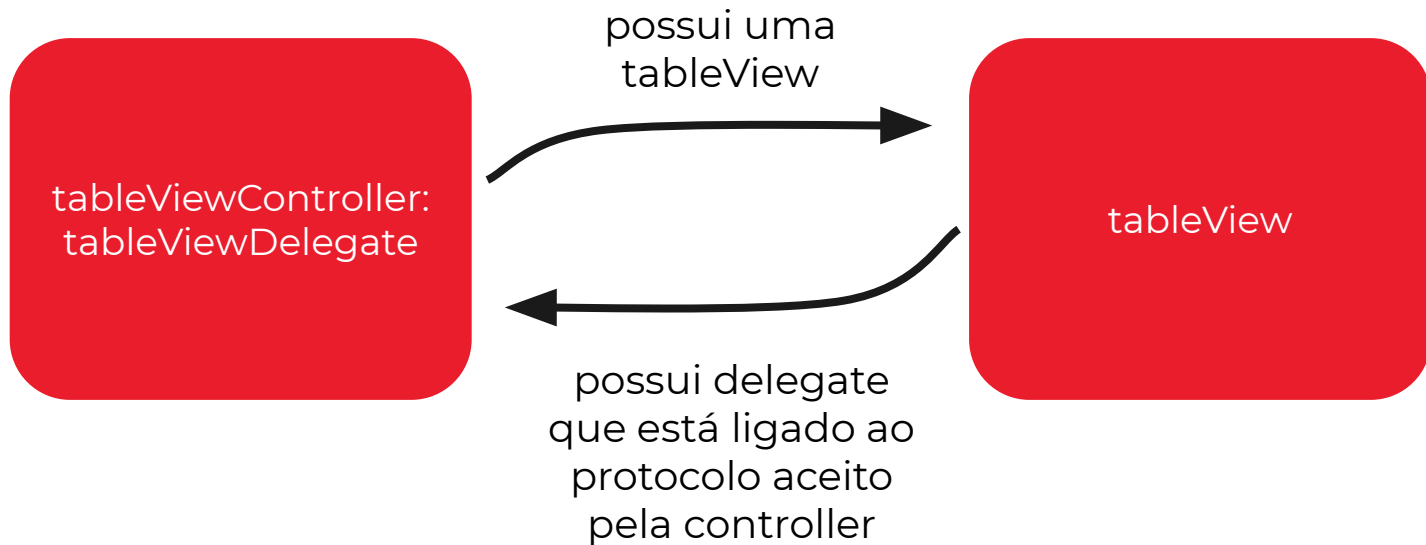
1. Protocolos



1. Protocolos



1. Protocolos



```
weak var delegate: UITableViewDelegate? { get set }
```


2. Closures

A TDC resolveu fazer um bot que dá oi pra todo mundo que chega e registra essa pessoa na ata:

2. Closures

A TDC resolveu fazer um bot que dá oi pra todo mundo que chega e registra essa pessoa na ata:

```
lazy var greet: Clousure = { attendee in  
    print("Hey! A new attendee! Welcome \((attendee.name)!")  
    self.attendee = attendee  
}
```

```
tdcSP.greet(you)
```

2. Closures

A TDC resolveu fazer um bot que dá oi pra todo mundo que chega e registra essa pessoa na ata:

```
lazy var greet: Clousure = { attendee in  
    print("Hey! A new attendee! Welcome \((attendee.name)!")  
    self.attendee = attendee  
}
```

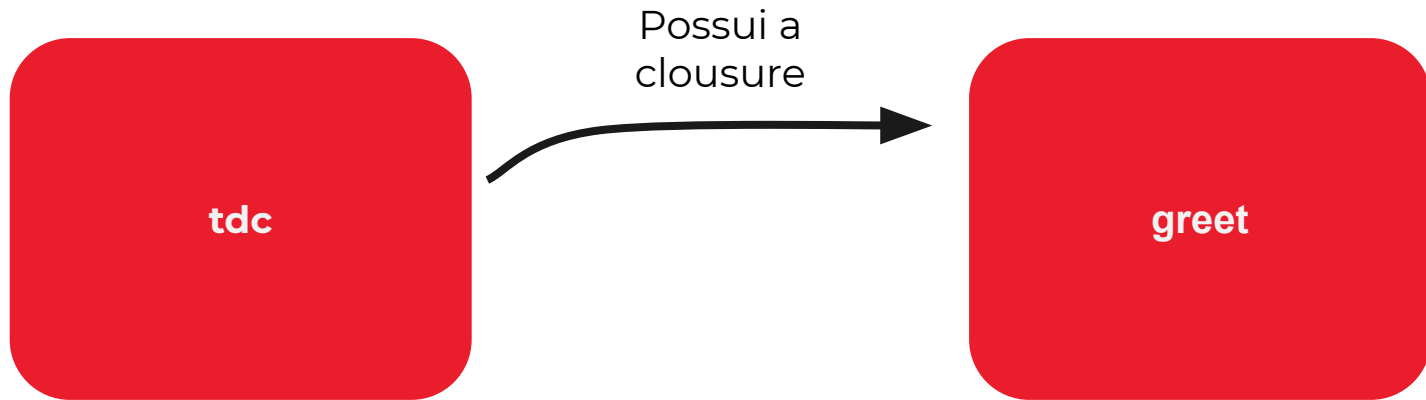


```
tdcSP.greet(you)
```

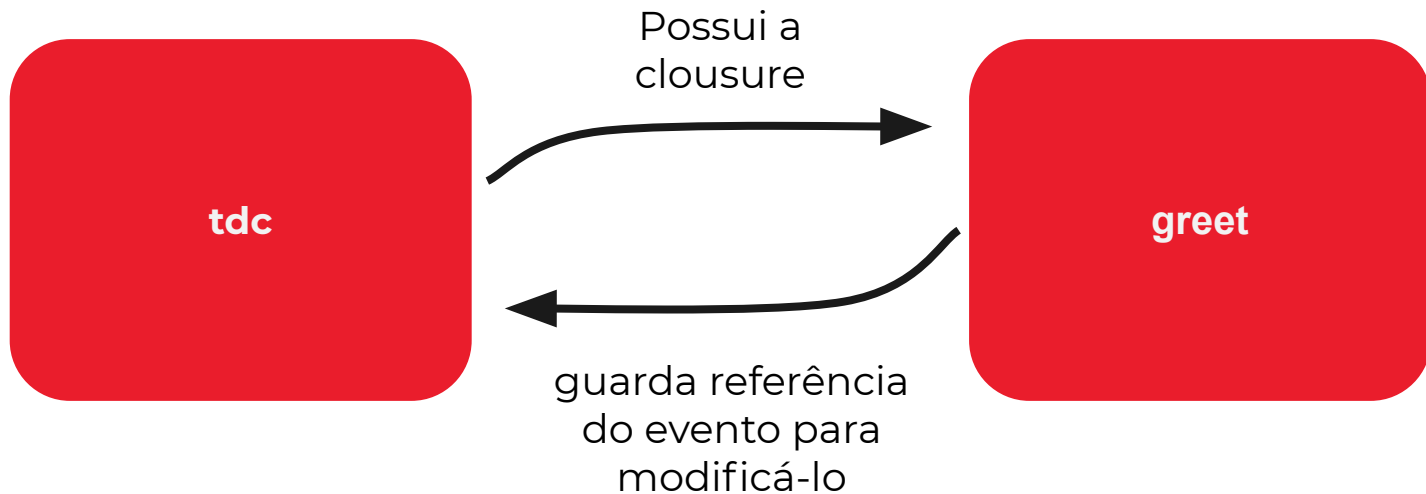
ATENÇÃO!!!

Bem aqui a clousure está **alterando self**. Por isso ela precisa guardar o caminho da memória desse objeto, se não ele não vai conseguir ter acesso.

2. Closures



2. Closures



2. Closures

```
lazy var greet: Closure = { attendee in  
    print("Hey! A new attendee! Welcome \ (attendee.name)!")  
    self.attendee = attendee  
}
```

2. Closures

Isso o Xcode não mostra



```
lazy var greet: Closure = { [strong self] attendee in  
    print("Hey! A new attendee! Welcome \(attendee.name)!")  
    self.attendee = attendee  
}
```

2. Closures

```
lazy var greet: Closure = { [weak self] attendee in  
    print("Hey! A new attendee! Welcome \ (attendee.name)!")  
    self?.attendee = attendee  
}
```


2. Closures

```
lazy var greet: Closure = { [weak self] attendee in  
    print("Hey! A new attendee! Welcome \ (attendee.name)!")  
    self?.attendee = attendee  
}
```





**Mas como sei
que meu app
tem isso?**

I'M A LOVER ❤️ REVOLUCIONAR O UNIVERSO DA ALIMENTAÇÃO

Xcode



Ou
(e)

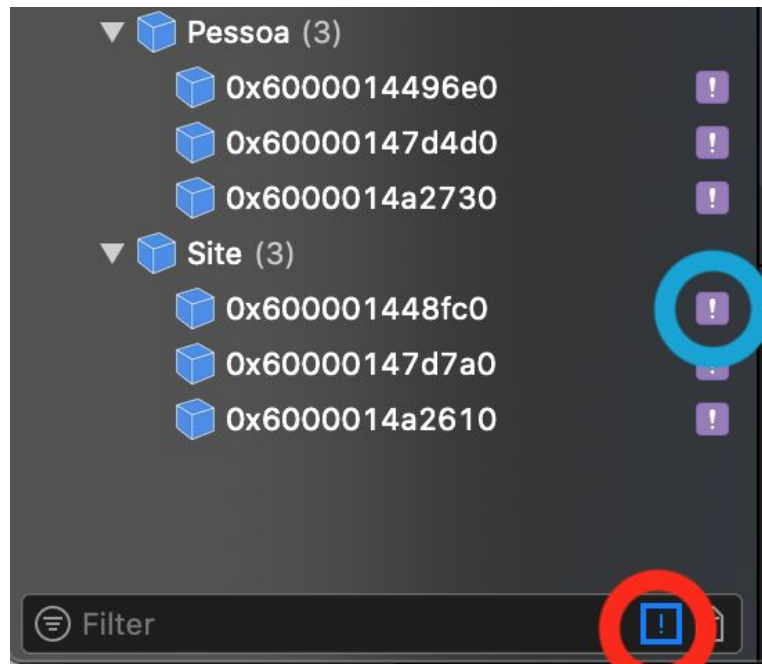
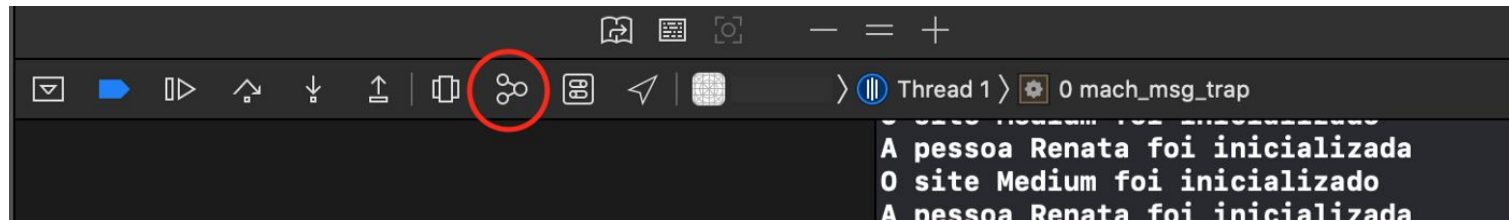
Instruments



Xcode



(demo)



Dentro do círculo azul está um exemplo de warning. Ela avisa que ali tem problemas de memória, se você clicar nela, terá mais detalhes.

Dentro do círculo vermelho temos um filtro que mostra apenas referências que possuem warnings, pressionando ele fica bem mais fácil ver onde tem problemas.



Pessoa



Site

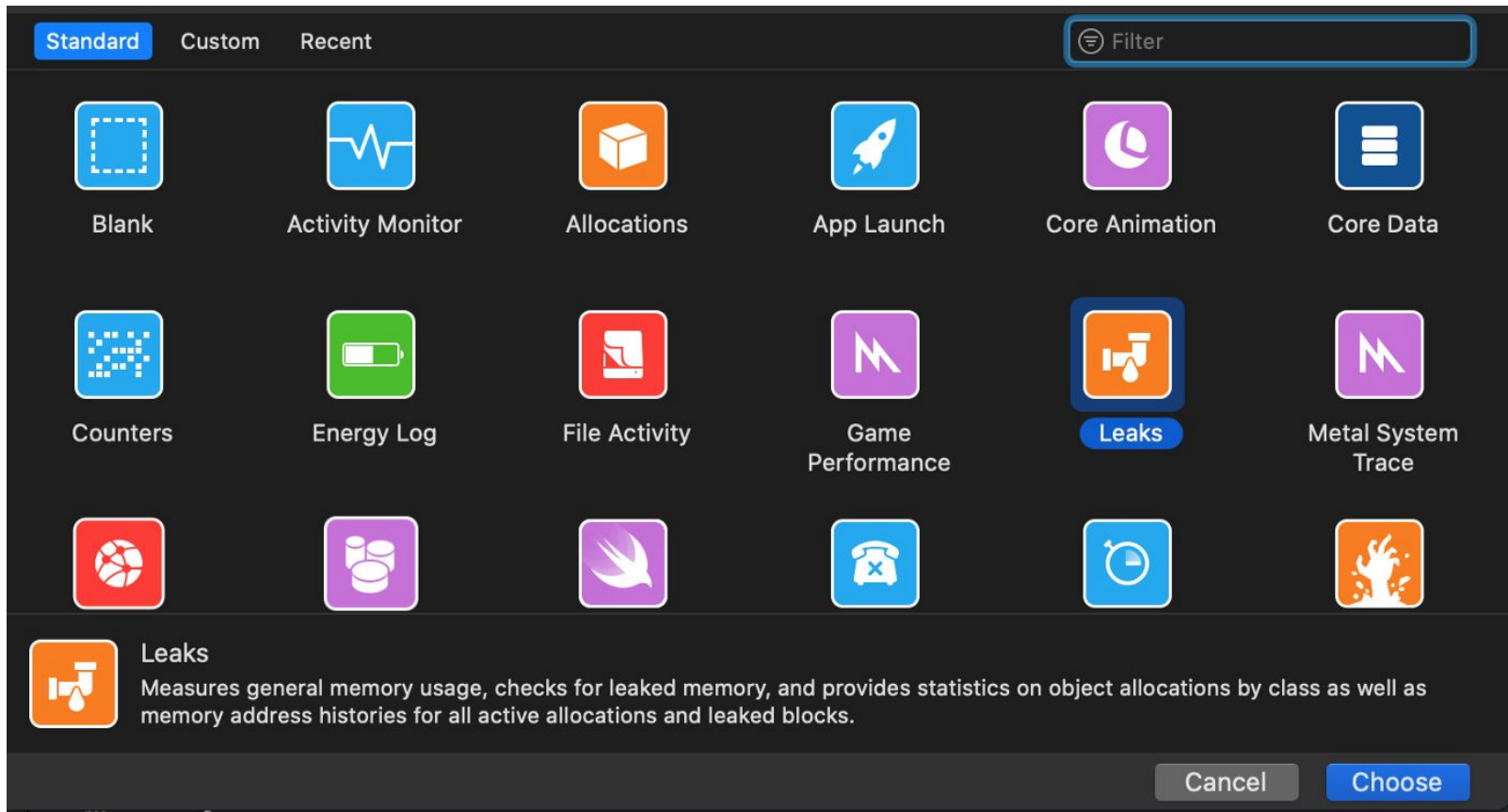
Instruments

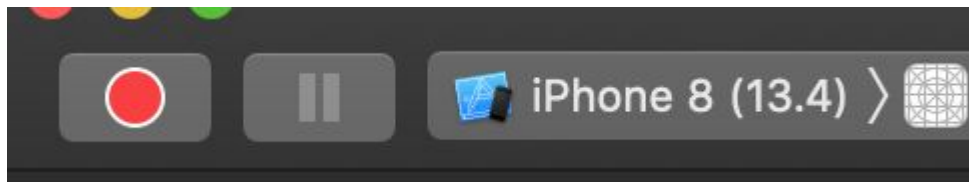


(demo)

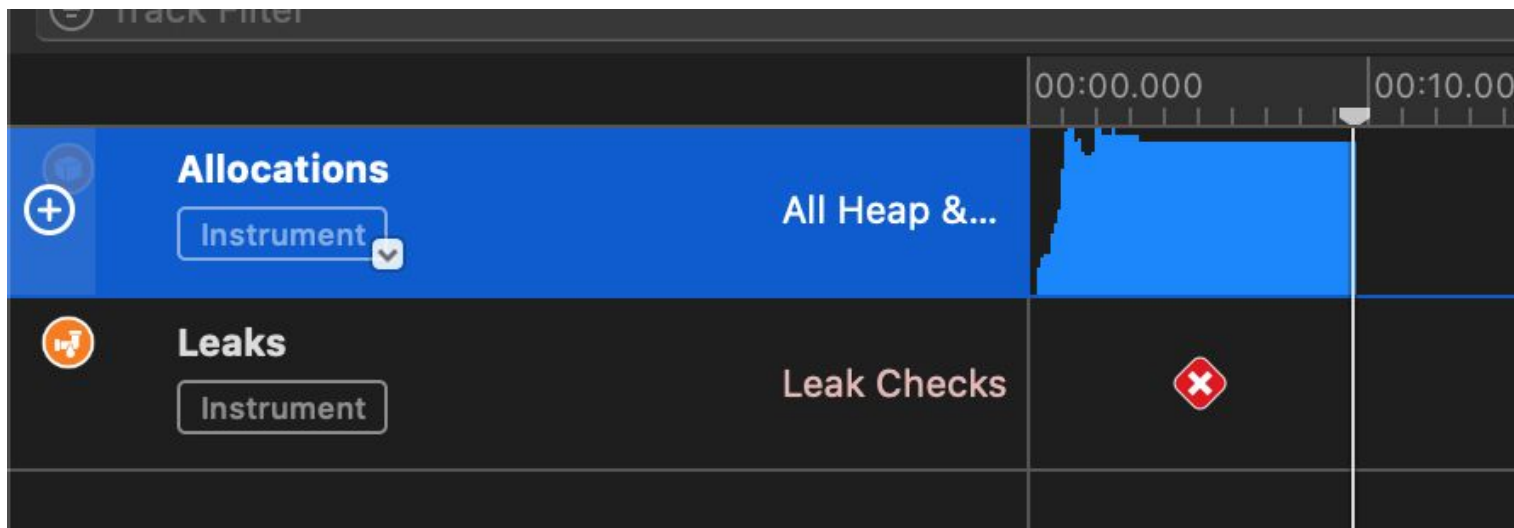


I'M A LOVER ❤️ REVOLUCIONAR O UNIVERSO DA ALIMENTAÇÃO





I'M A LOVER ❤️ REVOLUCIONAR O UNIVERSO DA ALIMENTAÇÃO





O verde indica que não existem leaks

O vermelho indica que existem novos leaks

O cinza indica que existem leaks, mas eles já foram descobertos antes

Leaked Object				
#				
Address				
Size				
Resp				
▼ Pessoa	2	< multiple >	96 Bytes	
Pessoa	1	0x6000031a73c0	48 Bytes	
Pessoa	1	0x6000031b6be0	48 Bytes	
► Site	2	< multiple >	96 Bytes	

Código



bit.ly/cod-tdcsp

Texto



bit.ly/tdcsp20-gm

Referências

- ["Weak, Strong, Unowned, Oh My!" - A Guide to References in Swift](#)
- [Memory leaks by custom delegate](#)
- [Swift Closures Memory Hell in iOS Development - Alem Utemissov](#)
- [Memory Leaks in Swift. Unit Testing and other tools to avoid... | by Leandro Pérez | Flawless iOS](#)

dúvidas?
slido.com
#A451





#VemProiFood