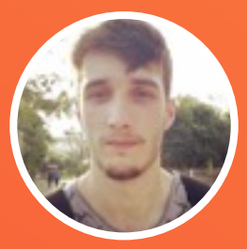
Swift: Gerenciamento de Memória



Txai Wieser

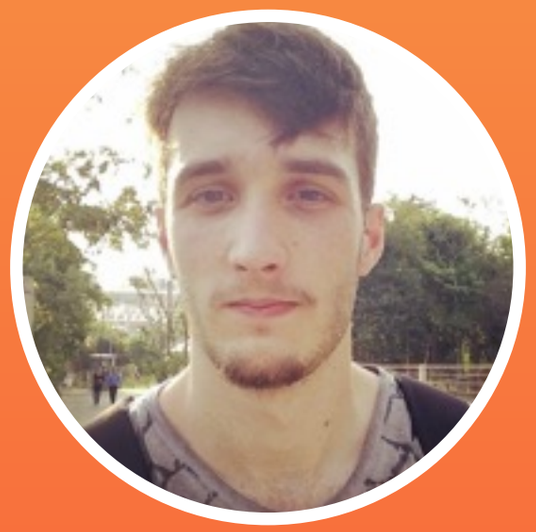
Repel

Recolorindo o Reino

Warren

FityIt

Ballphy



Txai Wieser

- Estudante de engenharia de computação na UFRGS.
- Desenvolvedor iOS

Swift: Gerenciamento de Memória

Gerenciador de Memória

MANUAL  AUTOMÁTICO

Gerenciador de Memória

MANUAL  AUTOMÁTICO

Programador

release

retain

alloc

Gerenciador de Memória

malloc

ObjectAlloc

heap

MallocScribble

MANUAL ⟷ AUTOMÁTICO

Programador

autorelease

__unsafe_unretained

free

release

autoreleasepool

__autoreleasing

destroy

retain

alloc

vmmap

assign

copy

Gerenciador de Memória

malloc

ObjectAlloc

heap

MallocScribble

MANUAL ⟷ AUTOMÁTICO

Programador

autorelease

__unsafe_unretained

free

release

autoreleasepool

__autoreleasing

destroy

retain

alloc

vmmap

assign

copy

Gerenciador de Memória

Programador

Garbage Collector

Gerenciador de Memória

MANUAL  AUTOMÁTICO

Programador

(-)trabalhoso

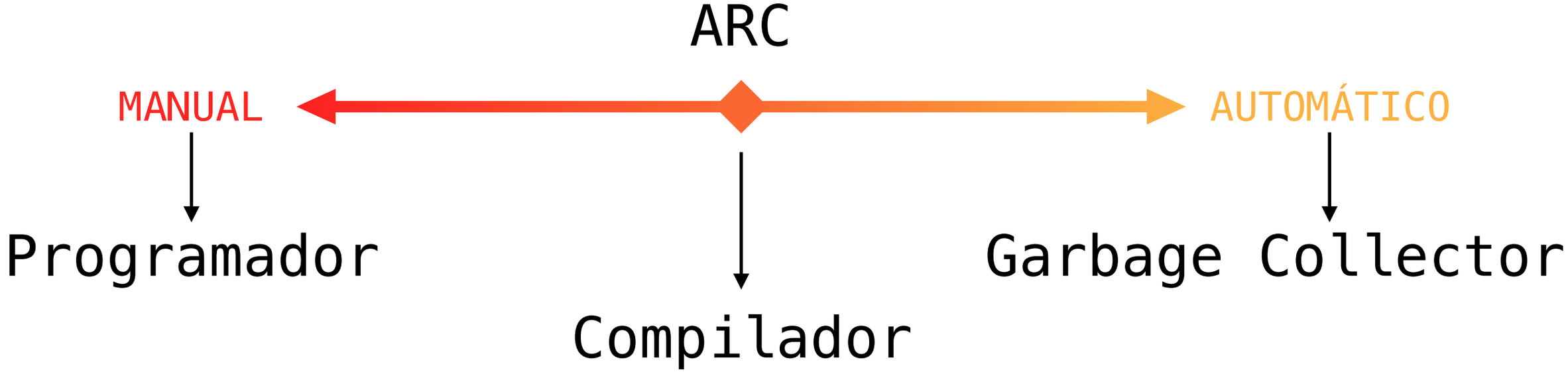
(+)determinístico

Garbage Collector

(-)afeta o runtime

(+)automático

Gerenciador de Memória



ARC

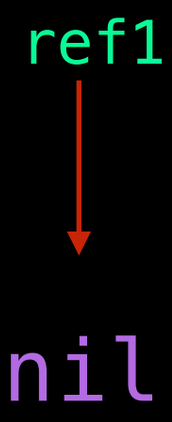
Reference counting is a technique of storing the number of references, pointers, or handles to a resource such as an object, block of memory, disk space or other resource. It may also refer, more specifically, to a garbage collection algorithm that uses these reference counts to deallocate objects which are no longer referenced.

Garbage Collector

Tracing garbage collection is a form of automatic memory management that consists of determining which objects should be deallocated by tracing which objects are reachable by a chain of references from certain “root” objects, and considering the rest as “garbage” and collecting them.

Automatic Reference Counting strong

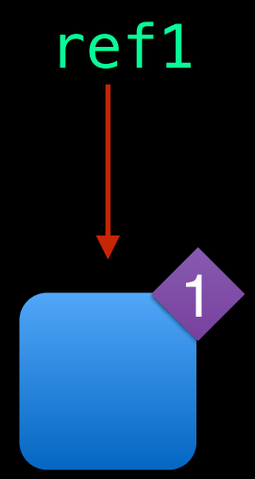
```
class Pessoa { }

var ref1: Pessoa?
```



Automatic Reference Counting

strong

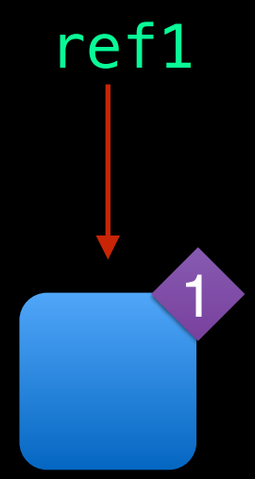
```
class Pessoa { }

var ref1: Pessoa?

ref1 = Pessoa()
```



Automatic Reference Counting

strong

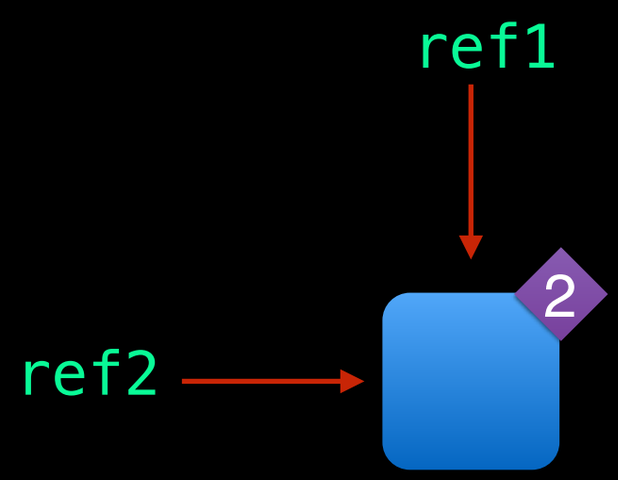
```
class Pessoa { }

var ref1: Pessoa?

ref1 = Pessoa()

var ref2: Pessoa?
```



Automatic Reference Counting

strong

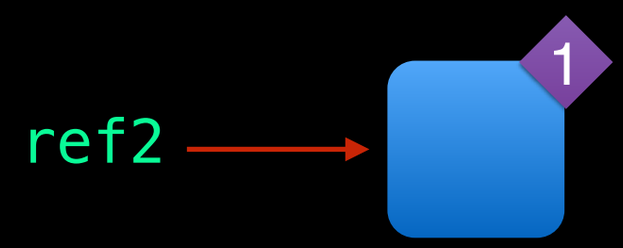
```
class Pessoa { }

var ref1: Pessoa?

ref1 = Pessoa()

var ref2: Pessoa?

ref2 = ref1
```



Automatic Reference Counting

strong

```
class Pessoa { }

var ref1: Pessoa?

ref1 = Pessoa()

var ref2: Pessoa?

ref2 = ref1


ref1 = nil
```



Automatic Reference Counting

strong

```swift
class Pessoa { }

var ref1: Pessoa?

ref1 = Pessoa()

var ref2: Pessoa?

ref2 = ref1


ref1 = nil

ref2 = nil
```

Automatic Reference Counting

strong

```
class Pessoa { }

var ref1: Pessoa?
ref1 = Pessoa()

var ref2: Pessoa?
ref2 = ref1

ref1 = nil

ref2 = nil
```

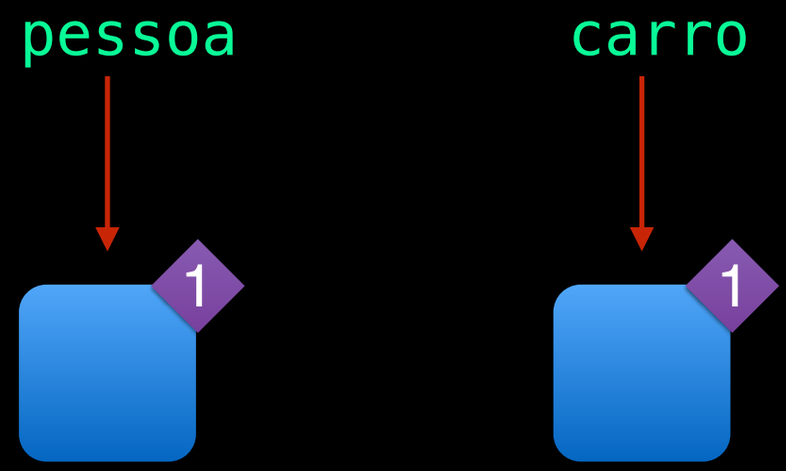
ref1

ref2

Ciclos de Referências

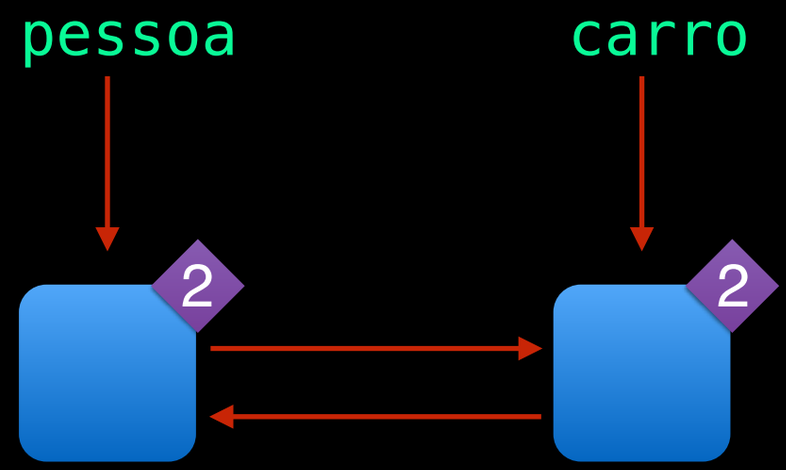
ARC (Vida Independente) weak

```
class Pessoa {
    var veículo: Carro?
}

class Carro {
    var proprietário:Pessoa?
}

var pessoa: Pessoa? = Pessoa()
var carro: Carro? = Carro()
```



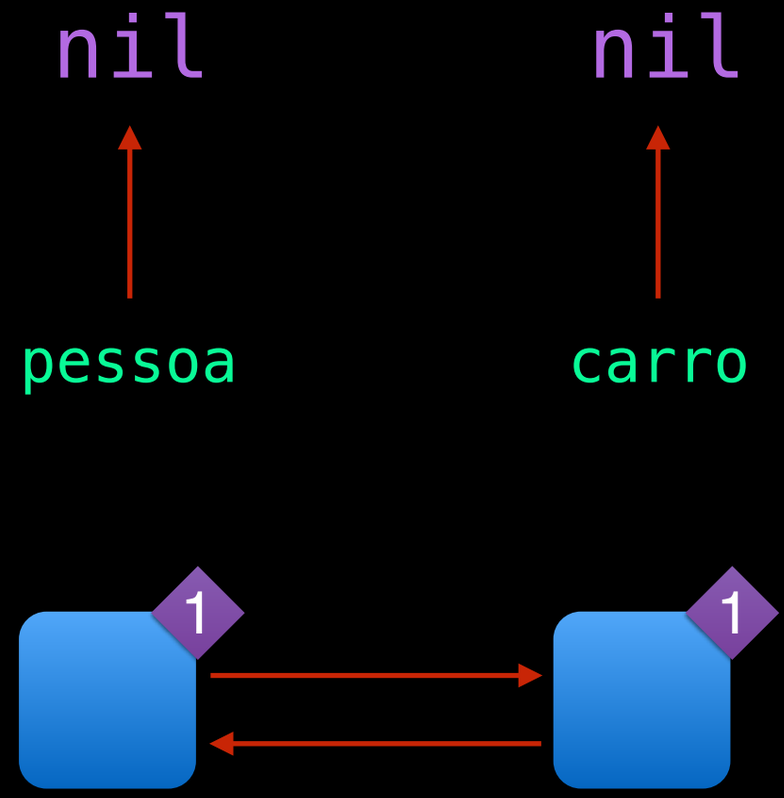
ARC (Vida Independente) weak

```
class Pessoa {
    var veículo: Carro?
}

class Carro {
    var proprietário:Pessoa?
}

var pessoa: Pessoa? = Pessoa()
var carro: Carro? = Carro()

pessoa!.veículo = carro
carro!.proprietário = pessoa
```



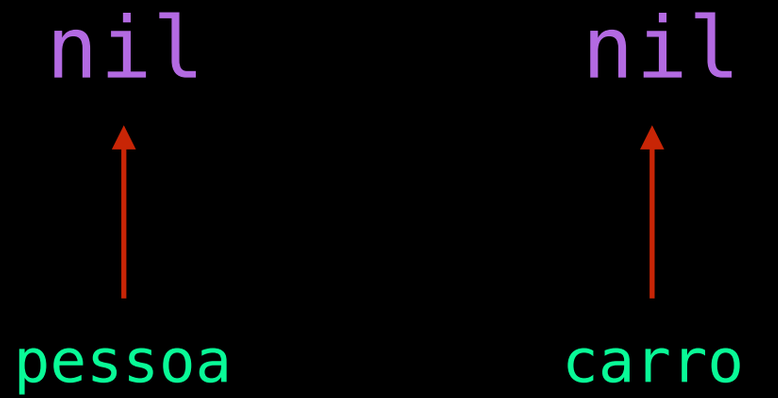
ARC (Vida Independente) weak

```
class Pessoa {
    var veículo: Carro?
}

class Carro {
    var proprietário:Pessoa?
}

var pessoa: Pessoa? = Pessoa()
var carro: Carro? = Carro()

pessoa!.veículo = carro
carro!.proprietário = pessoa


carro = nil
pessoa = nil
```



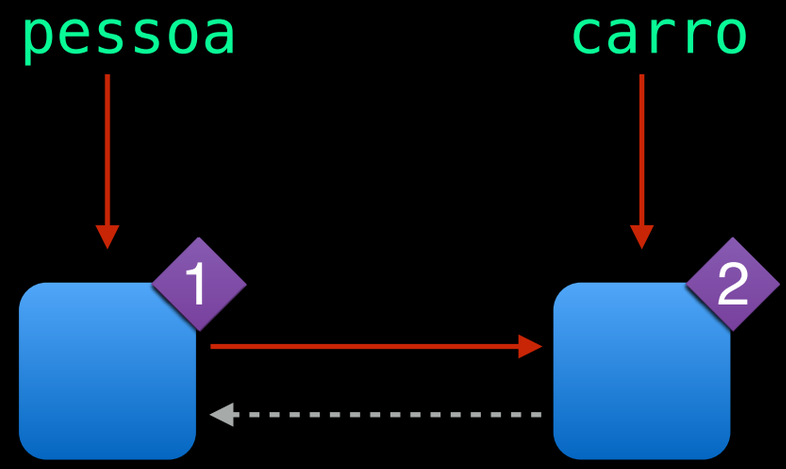
ARC (Vida Independente) weak

```
class Pessoa {
    var veículo: Carro?
}

class Carro {
    var proprietário:Pessoa?
}

var pessoa: Pessoa? = Pessoa()
var carro: Carro? = Carro()

pessoa!.veículo = carro
carro!.proprietário = pessoa


carro = nil
pessoa = nil
```



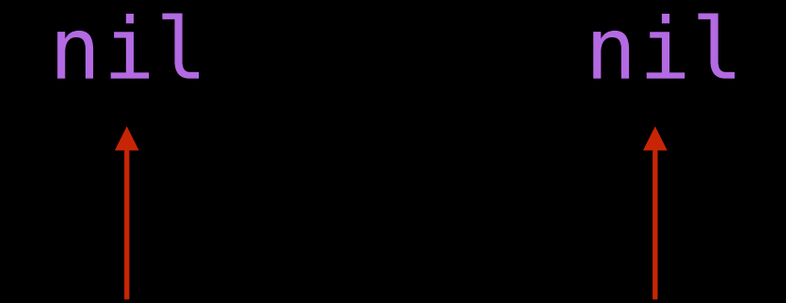
ARC (Vida Independente) weak

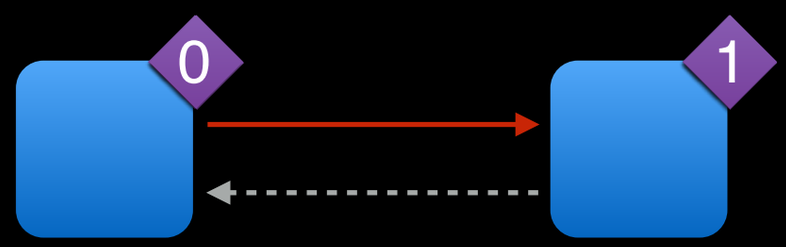
```
class Pessoa {
    var veículo: Carro?
}

class Carro {
    weak var proprietário:Pessoa?
}

var pessoa: Pessoa? = Pessoa()
var carro: Carro? = Carro()

pessoa!.veículo = carro
carro!.proprietário = pessoa


carro = nil
pessoa = nil
```



ARC (Vida Independente) weak

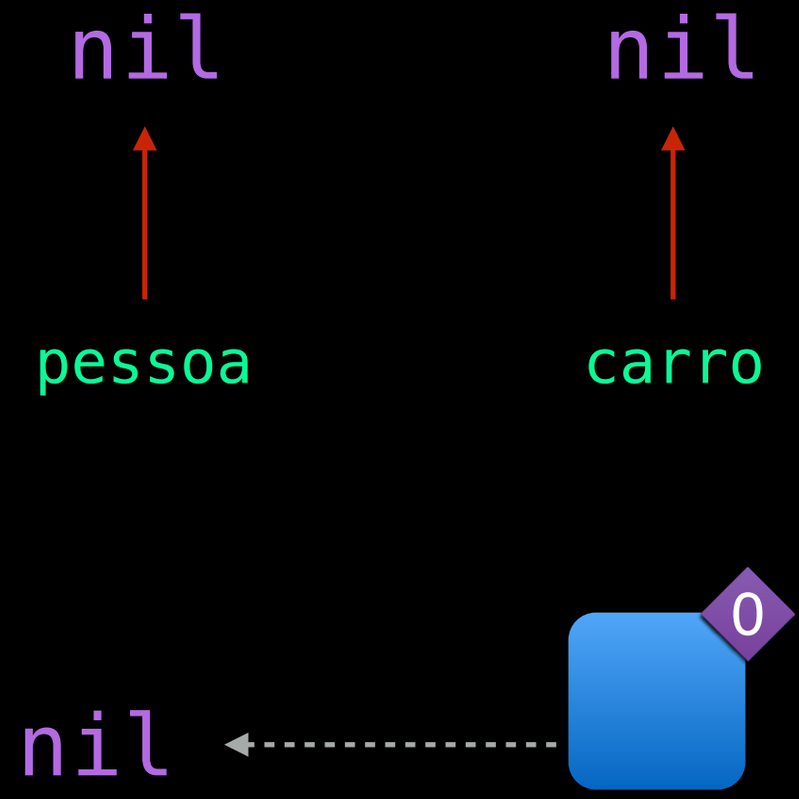
```
class Pessoa {
    var veículo: Carro?
}

class Carro {
    weak var proprietário:Pessoa?
}

var pessoa: Pessoa? = Pessoa()
var carro: Carro? = Carro()

pessoa!.veículo = carro
carro!.proprietário = pessoa


carro = nil
pessoa = nil
```





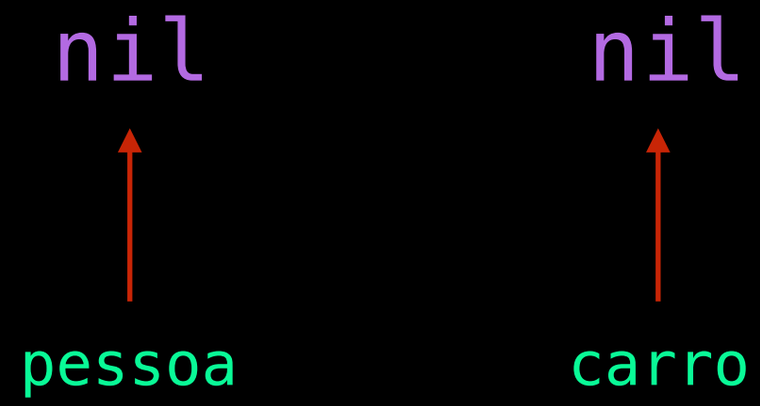
ARC (Vida Independente) weak

```
class Pessoa {
    var veículo: Carro?
}

class Carro {
    weak var proprietário:Pessoa?
}

var pessoa: Pessoa? = Pessoa()
var carro: Carro? = Carro()

pessoa!.veículo = carro
carro!.proprietário = pessoa


carro = nil
pessoa = nil
```



ARC (Vida Independente) weak

```
class Pessoa {
    var veículo: Carro?
}

class Carro {
    weak var proprietário:Pessoa?
}

var pessoa: Pessoa? = Pessoa()
var carro: Carro? = Carro()

pessoa!.veículo = carro
carro!.proprietário = pessoa


carro = nil
pessoa = nil
```

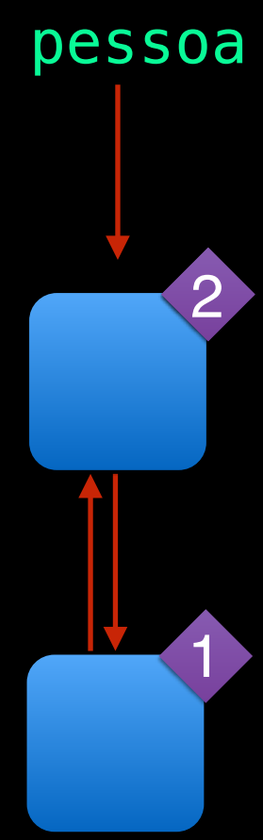


weak

(Vidas Independentes)

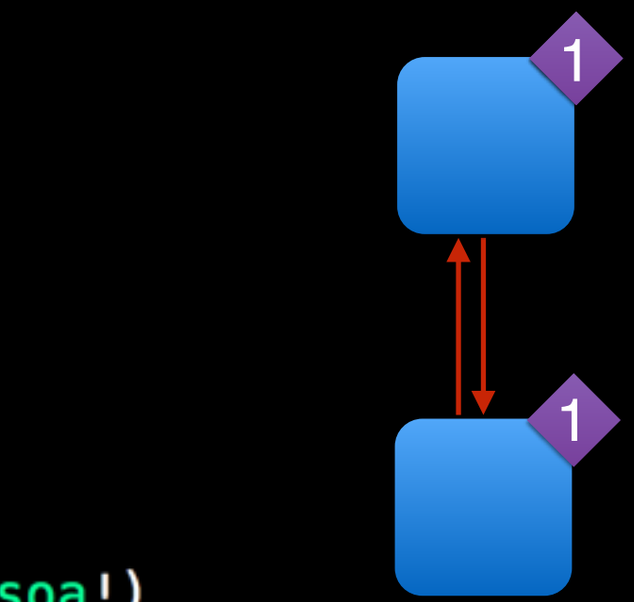
ARC (Vida Dependente) unowned

```swift
class Pessoa {
    var habilitação: Habilitação?
}

class Habilitação {
    let titular: Pessoa

    init(titular: Pessoa) {
        self.titular = titular
    }
}

var pessoa: Pessoa? = Pessoa()
pessoa!.habilitação = Habilitação(titular: pessoa!)
```



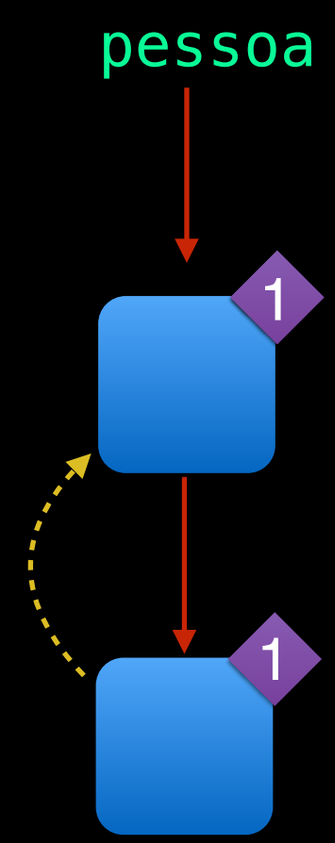
ARC (Vida Dependente) unowned

```swift
class Pessoa {
    var habilitação: Habilitação?
}

class Habilitação {
    let titular: Pessoa

    init(titular: Pessoa) {
        self.titular = titular
    }
}

var pessoa: Pessoa? = Pessoa()
pessoa!.habilitação = Habilitação(titular: pessoa!)

pessoa = nil
```



ARC (Vida Dependente) unowned

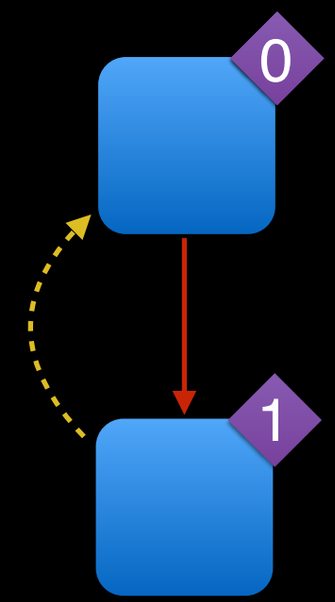
```swift
class Pessoa {
    var habilitação: Habilitação?
}

class Habilitação {
    unowned let titular: Pessoa

    init(titular: Pessoa) {
        self.titular = titular
    }
}

var pessoa: Pessoa? = Pessoa()
pessoa!.habilitação = Habilitação(titular: pessoa!)
```



ARC (Vida Dependente) unowned

```swift
class Pessoa {
    var habilitação: Habilitação?
}

class Habilitação {
    unowned let titular: Pessoa

    init(titular: Pessoa) {
        self.titular = titular
    }
}

var pessoa: Pessoa? = Pessoa()
pessoa!.habilitação = Habilitação(titular: pessoa!)

pessoa = nil
```

nil ← pessoa

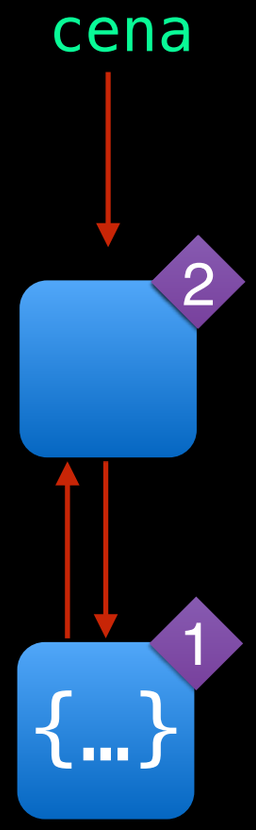


unowned

(Vida Dependente)

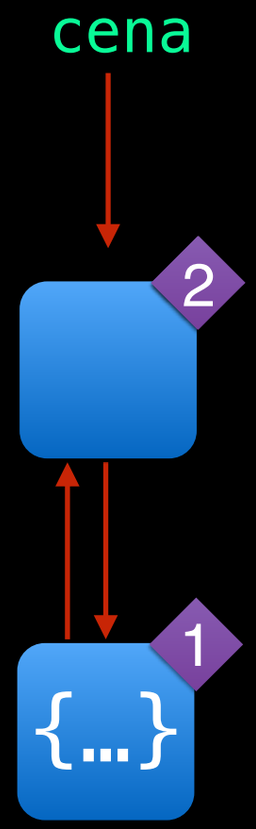
Closures

```
class Botão {
    var action: (()->())?
}

class Cena {
    let botão = Botão()
    var contagem = 0

    init() {
        botão.action = {
            contagem += 1
        }
    }
}


var cena:Cena? = Cena()
```



Closures

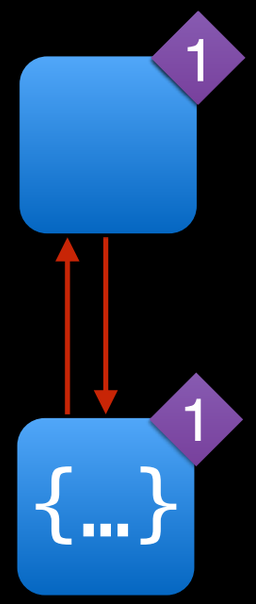
```
class Botão {
    var action: (()->())?
}

class Cena {
    let botão = Botão()
    var contagem = 0

    init() {
        botão.action = {
            self.contagem += 1
        }
    }
}

var cena:Cena? = Cena()
```



Closures

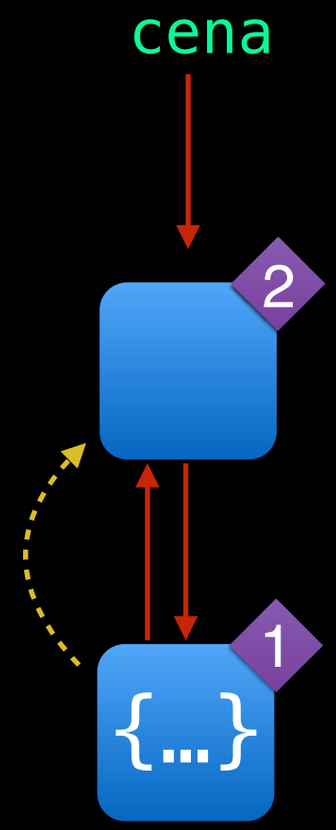
```
class Botão {
    var action: (()->())?
}

class Cena {
    let botão = Botão()
    var contagem = 0

    init() {
        botão.action = {
            self.contagem += 1
        }
    }
}


var cena:Cena? = Cena()

cena = nil
```

nil ← cena



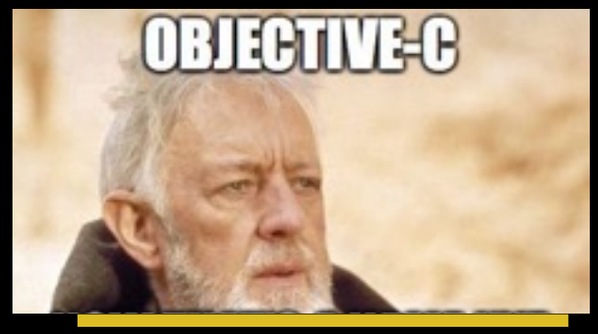
Closures

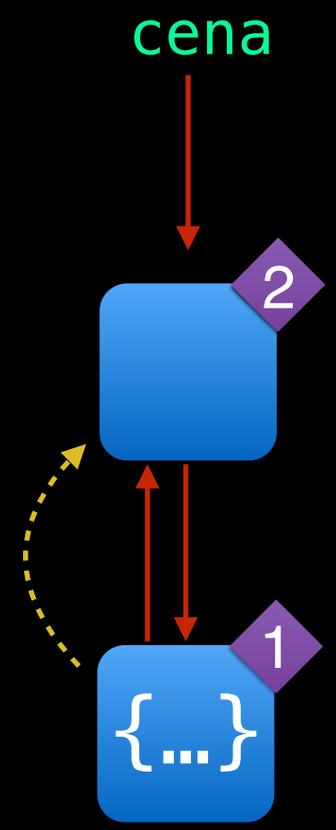
```
class Botão {
    var action: (()->())?
}

class Cena {
    let botão = Botão()
    var contagem = 0

    init() {
        self.botão.action = { [weak self] in
            self?.contagem += 1
        }
    }
}

var cena:Cena? = Cena()
```



Closures

```
class Botão {
    var action: (()->())?
}

class Cena {
    let botão = Botão()
    var contagem = 0

    init() {
        self.botão.action = { [weak self] in
            self?.contagem += 1
        }
    }
}


var cena:Cena? = Cena()
```





Closures

```
let closure: (()->())? = { [weak self] in
        if let strongSelf = self {
            strongSelf.view.alpha = 0.0
        }
    }
```

Closures

```
let resource = UIViewController()

let closure: (()->())? = { [weak self, weak resource] in
    if let strongSelf = self {
        strongSelf.view.alpha = 0.0
    }

    resource?.view.alpha = 0.0
}
```

Closures

(Lista de Captura)

Prevenção

Cuidado ⚠️

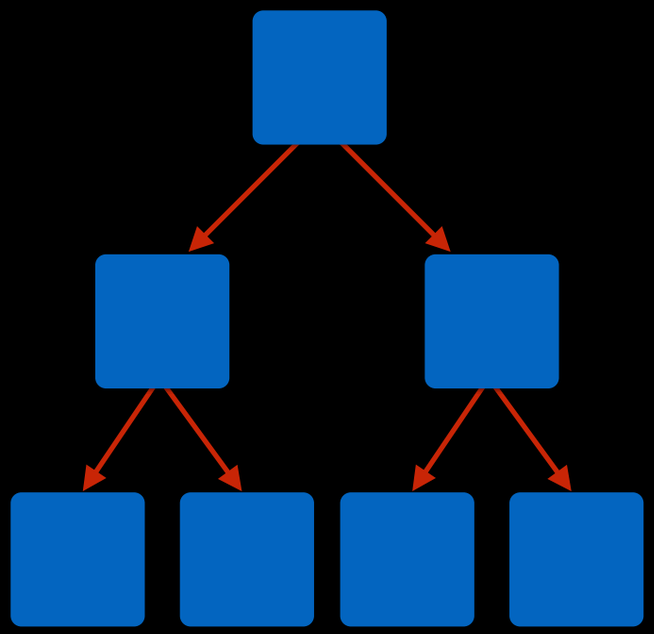
- Relações de Posse
- Referencias para classes
- Closures
  - NSTimer
  - UIAlertController

Use logs:

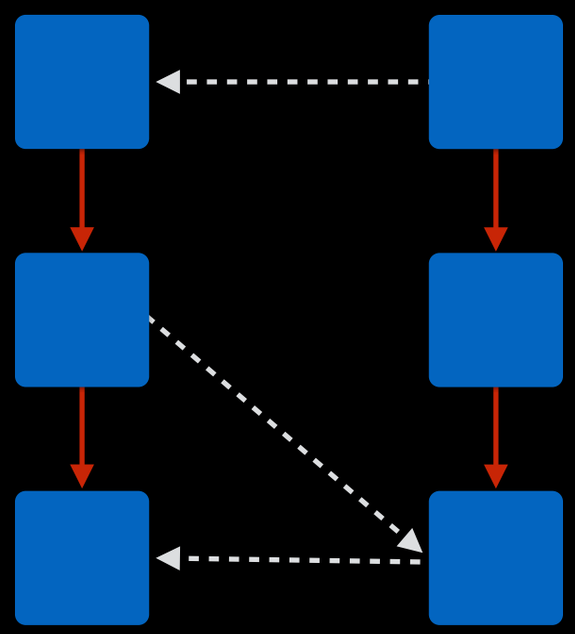
```
deinit {
    println("object is being deallocated")
}
```

```
-(void)dealloc
{
       NSLog(@"object is being deallocated");
}
```

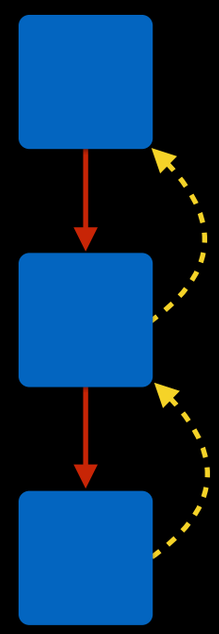
strong



weak



unowned



Value Types

Reference Types

debugging

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help Qui 09 57 16 78%

ARC › iPhone 6 | Finished running - Profiling ARC.app on iPhone 6

ARC › ARC › ViewController.swift › M viewDidLoad()

- ARC
  - 2 targets, iOS SDK 8.3
  - ARC
    - AppDelegate.swift
    - ViewController.swift
    - Main.storyboard
    - Images.xcassets
    - LaunchScreen.xib
    - Supporting Files
  - ARCTests
  - Products

```swift
//
//  ViewController.swift
//  ARC
//
//  Created by Txai Wieser on 26/03/15.
//  Copyright (c) 2015 Txai Wieser. All rights reserved.
//

import UIKit


class Pessoa {
    var veículo:Carro?
}

class Carro {
    var proprietário:Pessoa?
}


class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        var pessoa:Pessoa? = Pessoa()
        var carro:Carro? = Carro()

        pessoa!.veículo = carro
        carro!.proprietário = pessoa
```

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help Qui 09:57:35 78%

About Xcode

Preferences... ⌘,
Behaviors ▶

Open Developer Tool ▶
Services ▶

Hide Xcode ⌘H
Hide Others ⌥⌘H
Show All

Quit Xcode ⌘Q

Instruments
iOS Simulator
Accessibility Inspector
FileMerge
Application Loader

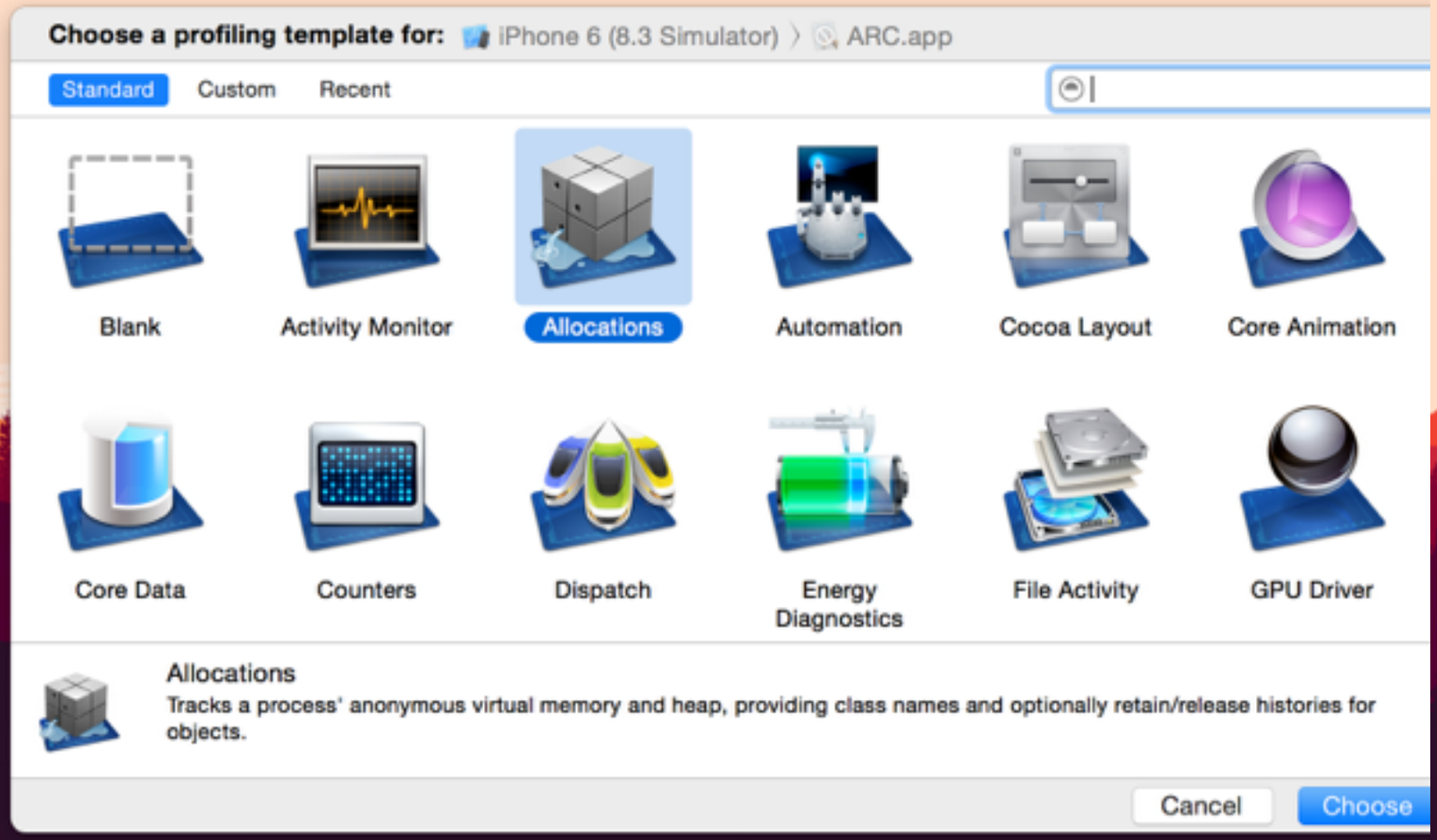
More Developer Tools...

shed running - Profiling ARC.app on iPhone 6

ARC 〉 ARC 〉 ViewController.swift 〉 M viewDidLoad()

AppDelegate.swift
ViewController.swift
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
ARCTests
Products

```
iewController.swift
RC
//
//  Created by Txai Wieser on 26/03/15.
//  Copyright (c) 2015 Txai Wieser. All rights reserved.
//

import UIKit


class Pessoa {
    var veículo:Carro?
}

class Carro {
    var proprietário:Pessoa?
}


class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        var pessoa:Pessoa? = Pessoa()
        var carro:Carro? = Carro()

        pessoa!.veículo = carro
        carro!.proprietário = pessoa
```


Choose a profiling template for: iPhone 6 (8.3 Simulator) 〉 ARC.app
Standard
Custom
Recent
Blank
Activity Monitor
Allocations
Automation
Cocoa Layout
Core Animation
Core Data
Counters
Dispatch
Energy Diagnostics
File Activity
GPU Driver
Allocations
Tracks a process' anonymous virtual memory and heap, providing class names and optionally retain/release histories for objects.
Cancel
Choose

Instruments File Edit View Instrument Window Help Qui 10 06 32 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app | Run 0 of 0 | 00:00:00

Instruments

Allocations — Analyzes the memory life-cycle of process' allocated blocks; can record reference counting events.

VM Tracker — Tracks the virtual memory space of a process over time, identifying regions by tag and reporting usage statistics.

Allocations 〉 Statistics 〉 Allocation Summary | Instrument Detail

Graph	Category	Persistent By…	# Persistent	# Transient	Total Bytes	# Total	Transient/Total Bytes

Launch Configuration for All Allocations

- [x] Discard unrecorded data upon stop
- [ ] Discard events for freed memory
- [ ] Only track VM allocations

Launch Configuration for Heap Allocations

- [ ] Record reference counts
- [x] Identify virtual C++ objects
- [ ] Enable NSZombie detection

Track Display

Style: Current Bytes

Type: Overlay

Recorded Types

	Type String	Search	Action
☑	*	is Contai…	Record
☐	NS	is Prefix	Ignore
☐	CF	is Prefix	Ignore
☐	Malloc	is Prefix	Ignore

+ −

Rules are evaluated top-to-bottom; later rules override earlier ones.

Instruments File Edit View Instrument Window Help Qui 10:06:37 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app | Run 1 of 1 | 00:00:03

Instruments

Allocations

VM Tracker

Allocations 〉 Statistics 〉 Allocation Summary | Instrument Detail

Graph	Category	Persistent By...	# Persistent	# Transient	Total Bytes	# Total	Transient/Total Bytes
☑	All Heap & Anonymous VM	19,75 MB	20.200	200.887	32,51 MB	221.087	
☐	All Heap Allocations	3,67 MB	20.185	200.874	15,70 MB	221.059	
☐	All Anonymous VM	16,08 MB	15	13	16,80 MB	28	
☐	VM: Dispatch continuations	16,00 MB	1	0	16,00 MB	1	
☐	Malloc 2,00 MB	2,00 MB	1	0	2,00 MB	1	
☐	CFString (store)	262,22 KB	1.576	808	398,83 KB	2.384	
☐	Malloc 32 Bytes	176,94 KB	5.662	46.178	1,58 MB	51.840	
☐	Malloc 64 Bytes	113,19 KB	1.811	55.678	3,51 MB	57.489	
☐	CFString (immutable)	84,45 KB	2.277	7.147	667,66 KB	9.424	
☐	CFBasicHash (value-store)	82,08 KB	181	286	127,75 KB	467	
☐	CFBasicHash (key-store)	78,33 KB	140	204	116,88 KB	344	
☐	Malloc 4,00 KB	76,00 KB	19	309	1,28 MB	328	
☐	Malloc 16 Bytes	74,19 KB	4.748	23.660	443,88 KB	28.408	
☐	Malloc 512 Bytes	60,50 KB	121	54	87,50 KB	175	
☐	Malloc 128 Bytes	58,25 KB	466	590	132,00 KB	1.056	
☐	NSPathStore2	47,02 KB	132	1.296	383,58 KB	1.428	
☐	VM: CoreAnimation	40,00 KB	10	0	40,00 KB	10	
☐	Malloc 2,00 KB	38,00 KB	19	40	118,00 KB	59	
☐	Malloc 8,00 KB	32,00 KB	4	1	40,00 KB	5	
☐	Malloc 256 Bytes	30,25 KB	121	253	93,50 KB	374	
☐	Malloc 2,50 KB	25,00 KB	10	9	47,50 KB	19	
☐	Malloc 1,00 KB	24,00 KB	24	23	47,00 KB	47	
☐	Malloc 48 Bytes	22,69 KB	484	44.553	2,06 MB	45.037	
☐	VM: Allocation 20,00 KB	20,00 KB	1	0	20,00 KB	1	

Generation Analysis

Mark Generation

Allocation Lifespan

- All Allocations
- Created & Persistent
- Created & Destroyed

Allocation Type

- All Heap & Anonymous VM
- All Heap Allocations
- All VM Regions

Call Tree

- Separate by Category
- Separate by Thread
- Invert Call Tree
- Hide System Libraries
- Flatten Recursion

Call Tree Constraints

Count 0 ∞

Bytes -∞ ∞

Data Mining

Instruments File Edit View Instrument Window Help Qui 10:06:45 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app | Run 1 of 1 | 00:00:11

Instruments

Allocations

VM Tracker

Allocations 〉 Statistics 〉 Allocation Summary | pessoa

Graph	Category	Persistent By...	# Persistent	# Transient	Total Bytes	# Total	Transient/Total Bytes
☐	ARC.Pessoa	32 Bytes	1	0	32 Bytes	1	

Generation Analysis

Mark Generation

Allocation Lifespan

- ○ All Allocations
- ◉ Created & Persistent
- ○ Created & Destroyed

Allocation Type

- ◉ All Heap & Anonymous VM
- ○ All Heap Allocations
- ○ All VM Regions

Call Tree

- ☐ Separate by Category
- ☐ Separate by Thread
- ☐ Invert Call Tree
- ☐ Hide System Libraries
- ☐ Flatten Recursion

Call Tree Constraints

- ☐ Count 0 ∞
- ☐ Bytes -∞ ∞

Data Mining

Instruments File Edit View Instrument Window Help
Qui 10:06:45 75%
Instruments5
iPhone 6 (8.3 Simulator) ARC.app
Run 1 of 1 00:00:11
Instruments
Allocations
VM Tracker
Allocations Statistics Allocation Summary
pessoa
Graph Category Persistent By... # Persistent # Transient Total Bytes # Total Transient/Total Bytes
ARC.Pessoa 32 Bytes 1 0 32 Bytes 1
Generation Analysis
Mark Generation
Allocation Lifespan
All Allocations
Created & Persistent
Created & Destroyed
Allocation Type
All Heap & Anonymous VM
All Heap Allocations
All VM Regions
Call Tree
Separate by Category
Separate by Thread
Invert Call Tree
Hide System Libraries
Flatten Recursion
Call Tree Constraints
Count
Bytes
Data Mining

Instruments File Edit View Instrument Window Help Qui 10:06:45 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app | Run 1 of 1 | 00:00:11

Instruments

Allocations

VM Tracker

Allocations 〉 Statistics 〉 Allocation Summary | pessoa

Graph	Category	Persistent By...	# Persistent	# Transient	Total Bytes	# Total	Transient/Total Bytes
☐	ARC.Pessoa	32 Bytes	1	0	32 Bytes	1	

Generation Analysis

Mark Generation

Allocation Lifespan

- ○ All Allocations
- ◉ Created & Persistent
- ○ Created & Destroyed

Allocation Type

- ◉ All Heap & Anonymous VM
- ○ All Heap Allocations
- ○ All VM Regions

Call Tree

- ☐ Separate by Category
- ☐ Separate by Thread
- ☐ Invert Call Tree
- ☐ Hide System Libraries
- ☐ Flatten Recursion

Call Tree Constraints

- ☐ Count 0 ∞
- ☐ Bytes -∞ ∞

Data Mining

Instruments File Edit View Instrument Window Help Qui 10:06:45 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app | Run 1 of 1 | 00:00:11

Instruments

Allocations

VM Tracker

Allocations 〉 Statistics 〉 Allocation Summary | pessoa

Graph	Category	Persistent By...	# Persistent	# Transient	Total Bytes	# Total	Transient/Total Bytes
☐	ARC.Pessoa	32 Bytes	1	0	32 Bytes	1	

Generation Analysis

Mark Generation

Allocation Lifespan

- ○ All Allocations
- ◉ Created & Persistent
- ○ Created & Destroyed

Allocation Type

- ◉ All Heap & Anonymous VM
- ○ All Heap Allocations
- ○ All VM Regions

Call Tree

- ☐ Separate by Category
- ☐ Separate by Thread
- ☐ Invert Call Tree
- ☐ Hide System Libraries
- ☐ Flatten Recursion

Call Tree Constraints

- ☐ Count 0 ∞
- ☐ Bytes -∞ ∞

Data Mining

Instruments File Edit View Instrument Window Help Qui 10:06:45 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app | Run 1 of 1 | 00:00:11

Instruments

Allocations

VM Tracker

Allocations 〉 Statistics 〉 Allocation Summary | pessoa

Graph	Category	Persistent By...	# Persistent	# Transient	Total Bytes	# Total	Transient/Total Bytes
☐	ARC.Pessoa	32 Bytes	1	0	32 Bytes	1	

Generation Analysis

Mark Generation

Allocation Lifespan

- ○ All Allocations
- ◉ Created & Persistent
- ○ Created & Destroyed

Allocation Type

- ◉ All Heap & Anonymous VM
- ○ All Heap Allocations
- ○ All VM Regions

Call Tree

- ☐ Separate by Category
- ☐ Separate by Thread
- ☐ Invert Call Tree
- ☐ Hide System Libraries
- ☐ Flatten Recursion

Call Tree Constraints

- ☐ Count 0 ∞
- ☐ Bytes -∞ ∞

Data Mining

Instruments File Edit View Instrument Window Help Qui 10 06 52 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app | Run 1 of 1 | 00:00:18

Instruments

Allocations

VM Tracker

Allocations 〉 Statistics 〉 Allocation Summary | pessoa

Graph	Category	Persistent By...	# Persistent	# Transient	Total Bytes	# Total	Transient/Total Bytes
☐	ARC.Pessoa	32 Bytes	1	0	32 Bytes	1	

Generation Analysis

Mark Generation

Allocation Lifespan

- ○ All Allocations
- ◉ Created & Persistent
- ○ Created & Destroyed

Allocation Type

- ◉ All Heap & Anonymous VM
- ○ All Heap Allocations
- ○ All VM Regions

Call Tree

- ☐ Separate by Category
- ☐ Separate by Thread
- ☐ Invert Call Tree
- ☐ Hide System Libraries
- ☐ Flatten Recursion

Call Tree Constraints

- ☐ Count 0 ∞
- ☐ Bytes -∞ ∞

Data Mining

Instruments File Edit View Instrument Window Help Qui 10 06 52 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app | Run 1 of 1 | 00:00:18

Instruments

Allocations

VM Tracker

Allocations 〉 Statistics 〉 Allocation Summary | pessoa

Graph	Category	Persistent By...	# Persistent	# Transient	Total Bytes	# Total	Transient/Total Bytes
☐	ARC.Pessoa	32 Bytes	1	0	32 Bytes	1	

Generation Analysis

Mark Generation

Allocation Lifespan

- ○ All Allocations
- ◉ Created & Persistent
- ○ Created & Destroyed

Allocation Type

- ◉ All Heap & Anonymous VM
- ○ All Heap Allocations
- ○ All VM Regions

Call Tree

- ☐ Separate by Category
- ☐ Separate by Thread
- ☐ Invert Call Tree
- ☐ Hide System Libraries
- ☐ Flatten Recursion

Call Tree Constraints

- ☐ Count 0 ∞
- ☐ Bytes -∞ ∞

Data Mining

Instruments File Edit View Instrument Window Help Qui 10 06 52 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app | Run 1 of 1 | 00:00:18

Instruments

Allocations

VM Tracker

Allocations 〉 Statistics 〉 Allocation Summary | pessoa

Graph	Category	Persistent By...	# Persistent	# Transient	Total Bytes	# Total	Transient/Total Bytes
☐	ARC.Pessoa	32 Bytes	1	0	32 Bytes	1	

Generation Analysis

Mark Generation

Allocation Lifespan

- ○ All Allocations
- ◉ Created & Persistent
- ○ Created & Destroyed

Allocation Type

- ◉ All Heap & Anonymous VM
- ○ All Heap Allocations
- ○ All VM Regions

Call Tree

- ☐ Separate by Category
- ☐ Separate by Thread
- ☐ Invert Call Tree
- ☐ Hide System Libraries
- ☐ Flatten Recursion

Call Tree Constraints

- ☐ Count 0 ∞
- ☐ Bytes -∞ ∞

Data Mining

Instruments File Edit View Instrument Window Help Qui 10 07 06 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app | Run 1 of 1 | 00:00:32

Instruments

Allocations

VM Tracker

Allocations 〉 Statistics 〉 Allocation Summary 〉 ARC.Pessoa | pessoa

#	Address	Category	Timestamp	Live	Size	Responsible Library	Responsible Caller
0	0x7fc3d375...	ARC.Pessoa	00:01.025.762	•	32 Bytes	libswiftCore.dylib	swift_slowAlloc

Generation Analysis

Mark Generation

Allocation Lifespan

- All Allocations
- Created & Persistent
- Created & Destroyed

Allocation Type

- All Heap & Anonymous VM
- All Heap Allocations
- All VM Regions

Call Tree

- Separate by Category
- Separate by Thread
- Invert Call Tree
- Hide System Libraries
- Flatten Recursion

Call Tree Constraints

Count 0 ∞

Bytes -∞ ∞

Data Mining

Instruments File Edit View Instrument Window Help Qui 10 07 06 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app | Run 1 of 1 | 00:00:32

Instruments

Allocations

VM Tracker

Allocations 〉 Statistics 〉 Allocation Summary 〉 ARC.Pessoa | pessoa

#	Address	Category	Timestamp	Live	Size	Responsible Library	Responsible Caller
0	0x7fc3d375...	ARC.Pessoa	00:01.025.762	•	32 Bytes	libswiftCore.dylib	swift_slowAlloc

Generation Analysis

Mark Generation

Allocation Lifespan

- All Allocations
- Created & Persistent
- Created & Destroyed

Allocation Type

- All Heap & Anonymous VM
- All Heap Allocations
- All VM Regions

Call Tree

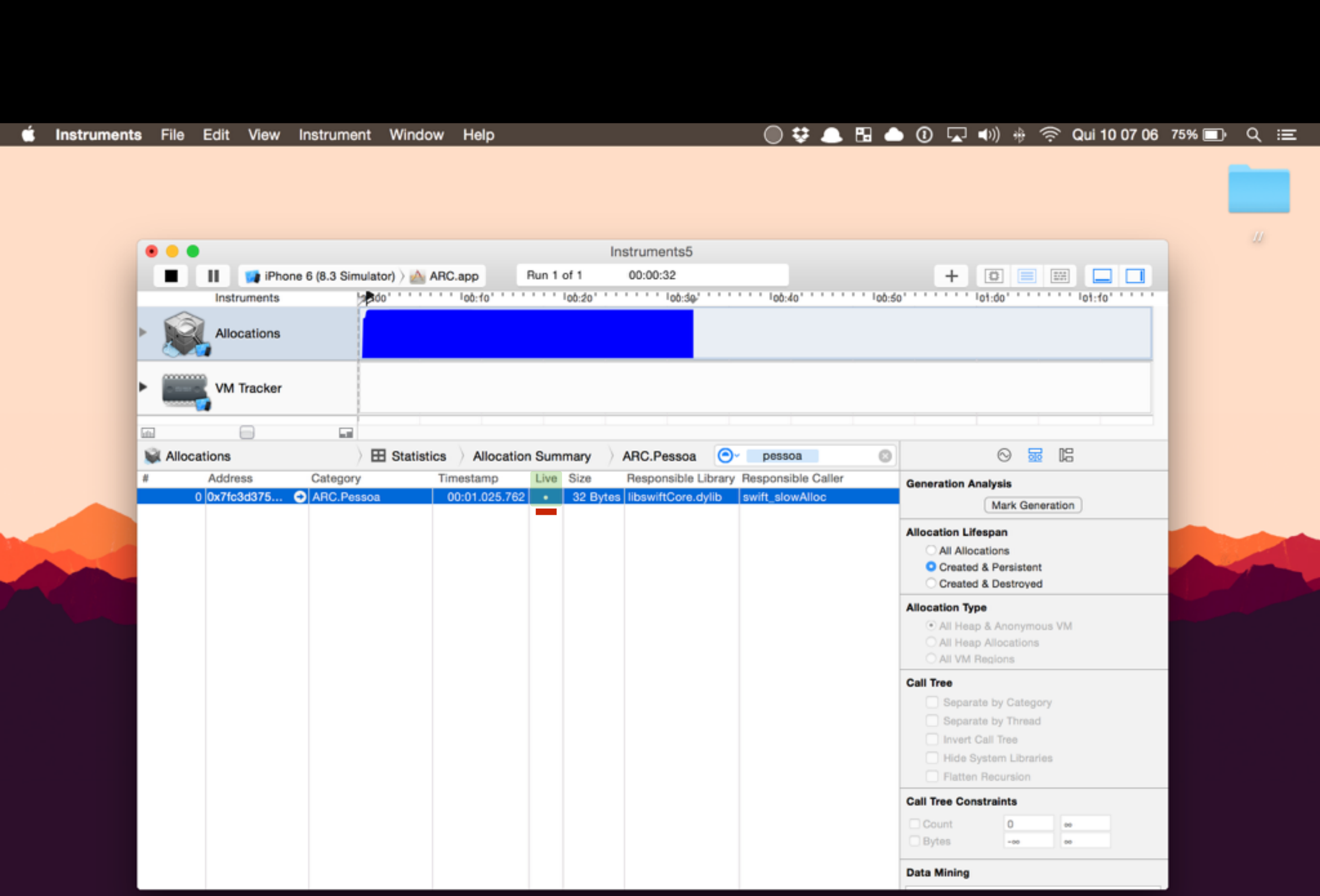
- Separate by Category
- Separate by Thread
- Invert Call Tree
- Hide System Libraries
- Flatten Recursion

Call Tree Constraints

Count 0 ∞

Bytes -∞ ∞

Data Mining


Instruments File Edit View Instrument Window Help
Qui 10 07 06
75%
Instruments5
iPhone 6 (8.3 Simulator)
ARC.app
Run 1 of 1
00:00:32
Instruments
Allocations
VM Tracker
Allocations
Statistics
Allocation Summary
ARC.Pessoa
pessoa
Address Category Timestamp Live Size Responsible Library Responsible Caller
0 0x7fc3d375... ARC.Pessoa 00:01.025.762 32 Bytes libswiftCore.dylib swift_slowAlloc
Generation Analysis
Mark Generation
Allocation Lifespan
All Allocations
Created & Persistent
Created & Destroyed
Allocation Type
All Heap & Anonymous VM
All Heap Allocations
All VM Regions
Call Tree
Separate by Category
Separate by Thread
Invert Call Tree
Hide System Libraries
Flatten Recursion
Call Tree Constraints
Count
Bytes
Data Mining

Instruments File Edit View Instrument Window Help Qui 10 07 00 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app | Run 1 of 1 | 00:00:26

Instruments

Allocations

VM Tracker

Allocations 〉 Statistics 〉 Allocation Summary 〉 ARC.Pessoa | pessoa

#	Address	Category	Timestamp	Live	Size	Responsible Library	Responsible Caller
0	0x7fc3d375...	ARC.Pessoa	00:01.025.762	•	32 Bytes	libswiftCore.dylib	swift_slowAlloc

Generation Analysis

Mark Generation

Allocation Lifespan

- All Allocations
- Created & Persistent
- Created & Destroyed

Allocation Type

- All Heap & Anonymous VM
- All Heap Allocations
- All VM Regions

Call Tree

- Separate by Category
- Separate by Thread
- Invert Call Tree
- Hide System Libraries
- Flatten Recursion

Call Tree Constraints

Count 0 ∞

Bytes -∞ ∞

Data Mining

Instruments File Edit View Instrument Window Help

Qui 10 07 00 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app

Run 1 of 1 00:00:26

Instruments

Allocations

VM Tracker

Allocations 〉 Statistics 〉 Allocation Summary 〉 ARC.Pessoa

pessoa

#	Address	Category	Timestamp	Live	Size	Responsible Library	Responsible Caller
0	0x7fc3d375...	ARC.Pessoa	00:01.025.762	•	32 Bytes	libswiftCore.dylib	swift_slowAlloc

Generation Analysis

Mark Generation

Allocation Lifespan

- All Allocations
- Created & Persistent
- Created & Destroyed

Allocation Type

- All Heap & Anonymous VM
- All Heap Allocations
- All VM Regions

Call Tree

- Separate by Category
- Separate by Thread
- Invert Call Tree
- Hide System Libraries
- Flatten Recursion

Call Tree Constraints

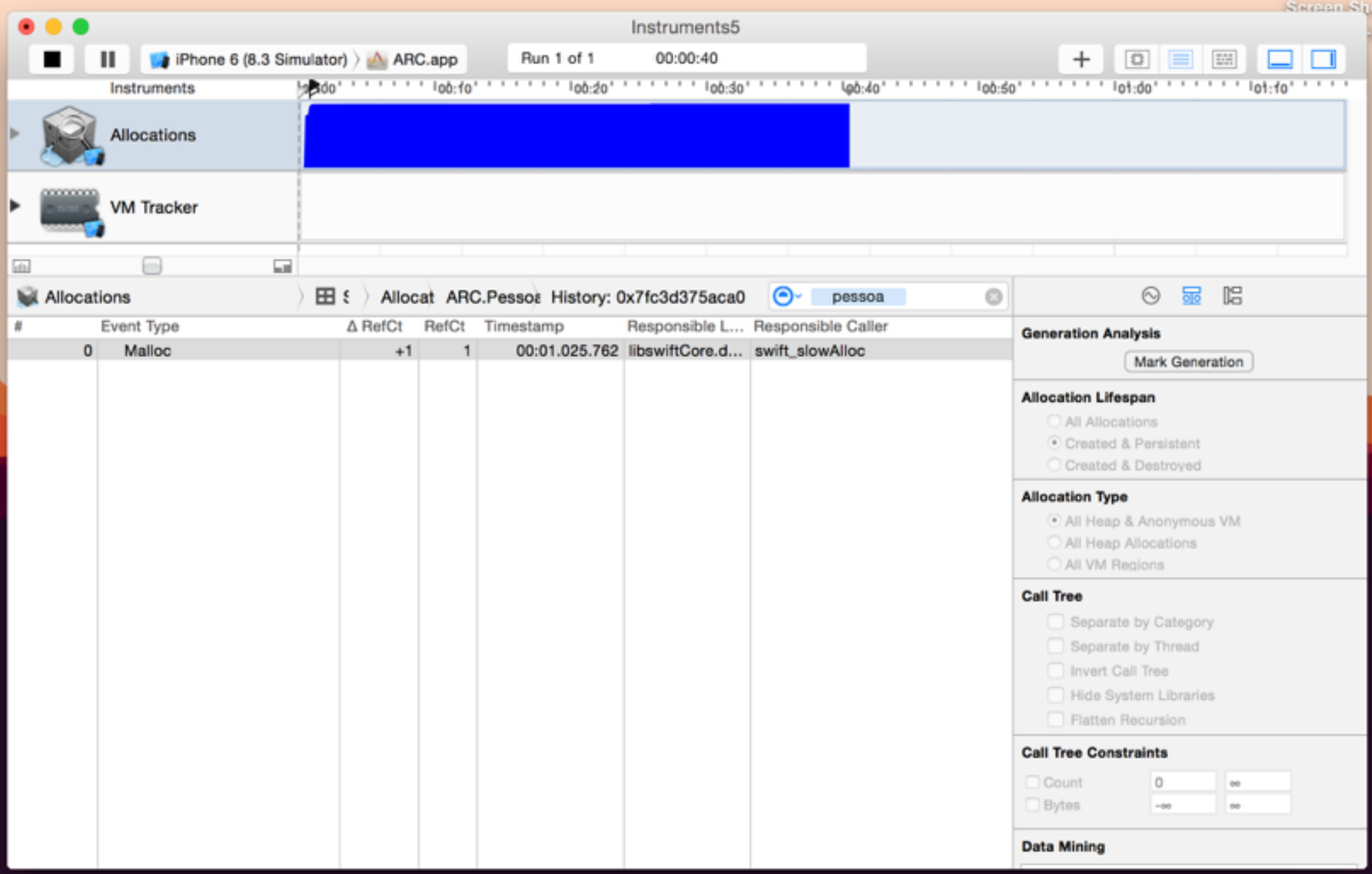
Count 0 ∞

Bytes -∞ ∞

Data Mining

Instruments File Edit View Instrument Window Help
Qui 10 07 00
75%
Instruments5
iPhone 6 (8.3 Simulator)
ARC.app
Run 1 of 1
00:00:26
Instruments
Allocations
VM Tracker
Allocations
Statistics
Allocation Summary
ARC.Pessoa
pessoa
Address Category Timestamp Live Size Responsible Library Responsible Caller
0 0x7fc3d375... ARC.Pessoa 00:01.025.762 • 32 Bytes libswiftCore.dylib swift_slowAlloc
Generation Analysis
Mark Generation
Allocation Lifespan
All Allocations
Created & Persistent
Created & Destroyed
Allocation Type
All Heap & Anonymous VM
All Heap Allocations
All VM Regions
Call Tree
Separate by Category
Separate by Thread
Invert Call Tree
Hide System Libraries
Flatten Recursion
Call Tree Constraints
Count
Bytes
Data Mining

Instruments File Edit View Instrument Window Help Qui 10 07 14 75%



Instruments File Edit View Instrument Window Help

Qui 10 07 14 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app

Run 1 of 1 00:00:40

Instruments

Allocations

VM Tracker

Allocations 〉 Allocat ARC.Pessoa History: 0x7fc3d375aca0

pessoa

#	Event Type	Δ RefCt	RefCt	Timestamp	Responsible L...	Responsible Caller
0	Malloc	+1	1	00:01.025.762	libswiftCore.d...	swift_slowAlloc

Generation Analysis

Mark Generation

Allocation Lifespan

- All Allocations
- Created & Persistent
- Created & Destroyed

Allocation Type

- All Heap & Anonymous VM
- All Heap Allocations
- All VM Regions

Call Tree

- Separate by Category
- Separate by Thread
- Invert Call Tree
- Hide System Libraries
- Flatten Recursion

Call Tree Constraints

- Count 0 ∞
- Bytes -∞ ∞

Data Mining

Instruments File Edit View Instrument Window Help

Qui 10 07 14 75%

Instruments5

iPhone 6 (8.3 Simulator) 〉 ARC.app

Run 1 of 1 00:00:40

Instruments

Allocations

VM Tracker

Allocations 〉 Allocat ARC.Pessoa History: 0x7fc3d375aca0

pessoa

#	Event Type	Δ RefCt	RefCt	Timestamp	Responsible L...	Responsible Caller
0	Malloc	+1	1	00:01.025.762	libswiftCore.d...	swift_slowAlloc

Generation Analysis

Mark Generation

Allocation Lifespan

- All Allocations
- Created & Persistent
- Created & Destroyed

Allocation Type

- All Heap & Anonymous VM
- All Heap Allocations
- All VM Regions

Call Tree

- Separate by Category
- Separate by Thread
- Invert Call Tree
- Hide System Libraries
- Flatten Recursion

Call Tree Constraints

- Count 0 ∞
- Bytes -∞ ∞

Data Mining

Referências:

- https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/AutomaticReferenceCounting.html
- http://www.ibm.com/developerworks/library/mo-ios-memory/

Thanks ;)

github.com/txaidw

twitter.com/txaidw