Introdução à Programação em Swift

Functional patterns

Protocols and extensions on structs

Pattern matching

Concise syntax

Closures

Generics

Fast iteration

Native collections

Optional types

Operator overloading

Object orientation

Namespaces

Tuples

Type inference

Clear mutability syntax

Read-Eval-Print-Loop (REPL)

Interactive playground

Multiple return types

Compile to native code

Chris Lattner

Por que  ?

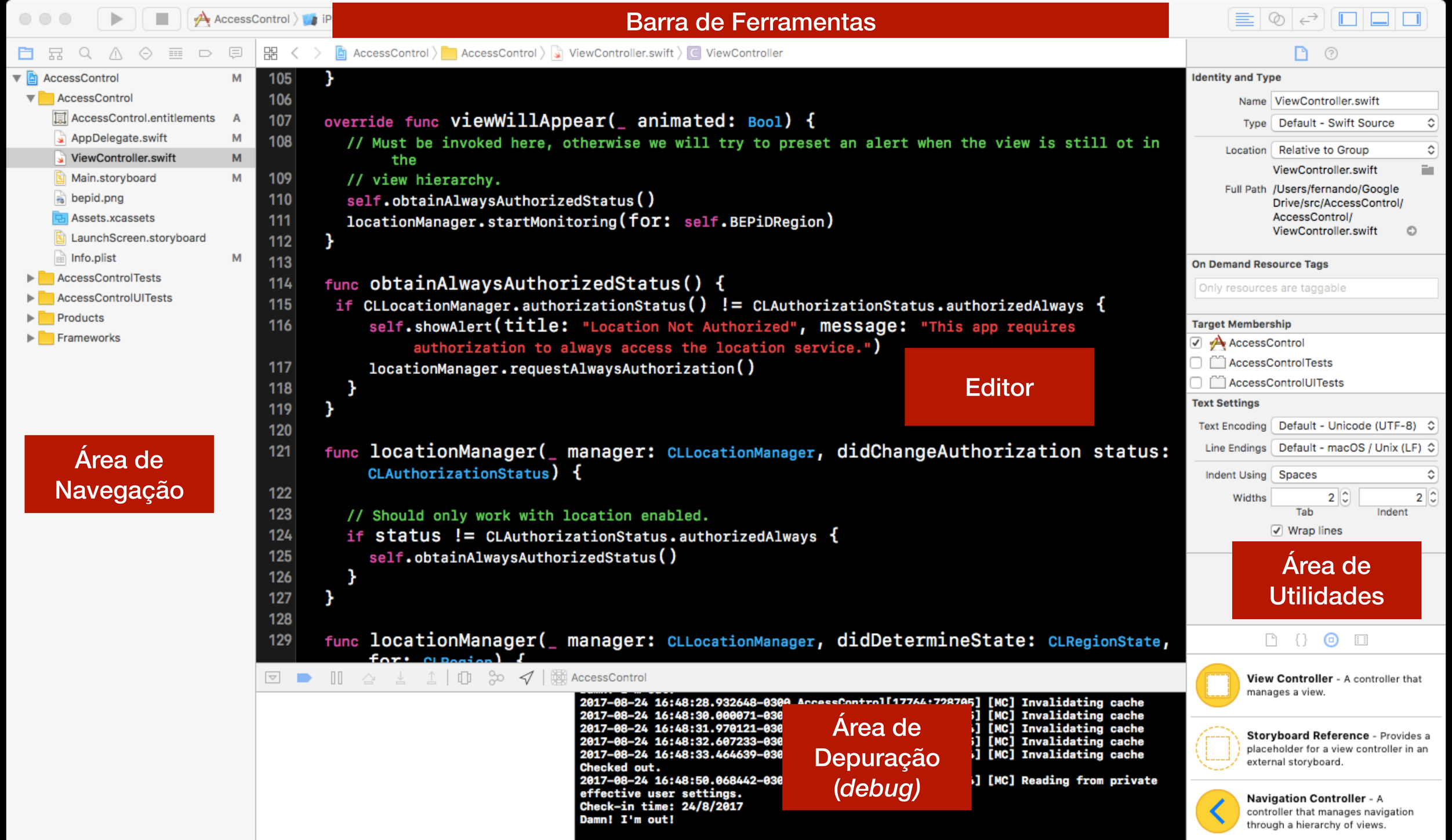
Sintaxe clara e inferência de tipos
Multiparadigma
Playgrounds
Integração com Objective-C
Software Livre

Há muita documentação sobre Swift:

https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/

https://swift.org/

A propósito, e Objective-C?


Barra de Ferramentas
AccessControl
AccessControl
ViewController.swift
ViewController
AccessControl
AccessControl.entitlements
AppDelegate.swift
ViewController.swift
Main.storyboard
bepid.png
Assets.xcassets
LaunchScreen.storyboard
Info.plist
AccessControlTests
AccessControlUITests
Products
Frameworks
Área de Navegação
105 }
106
107 override func viewWillAppear(_ animated: Bool) {
108 // Must be invoked here, otherwise we will try to preset an alert when the view is still ot in the
109 // view hierarchy.
110 self.obtainAlwaysAuthorizedStatus()
111 locationManager.startMonitoring(for: self.BEPiDRegion)
112 }
113
114 func obtainAlwaysAuthorizedStatus() {
115 if CLLocationManager.authorizationStatus() != CLAuthorizationStatus.authorizedAlways {
116 self.showAlert(title: "Location Not Authorized", message: "This app requires authorization to always access the location service.")
117 locationManager.requestAlwaysAuthorization()
118 }
119 }
120
121 func locationManager(_ manager: CLLocationManager, didChangeAuthorization status: CLAuthorizationStatus) {
122
123 // Should only work with location enabled.
124 if status != CLAuthorizationStatus.authorizedAlways {
125 self.obtainAlwaysAuthorizedStatus()
126 }
127 }
128
129 func locationManager(_ manager: CLLocationManager, didDetermineState: CLRegionState,
Editor
AccessControl
2017-08-24 16:48:28.932648-0300 AccessControl[17764:728705] [MC] Invalidating cache
2017-08-24 16:48:30.000071-030] [MC] Invalidating cache
2017-08-24 16:48:31.970121-030] [MC] Invalidating cache
2017-08-24 16:48:32.607233-030] [MC] Invalidating cache
2017-08-24 16:48:33.464639-030] [MC] Invalidating cache
Checked out.
2017-08-24 16:48:50.068442-030] [MC] Reading from private effective user settings.
Check-in time: 24/8/2017
Damn! I'm out!
Área de Depuração (debug)
Identity and Type
Name ViewController.swift
Type Default - Swift Source
Location Relative to Group
ViewController.swift
Full Path /Users/fernando/Google Drive/src/AccessControl/AccessControl/ViewController.swift
On Demand Resource Tags
Only resources are taggable
Target Membership
AccessControl
AccessControlTests
AccessControlUITests
Text Settings
Text Encoding Default - Unicode (UTF-8)
Line Endings Default - macOS / Unix (LF)
Indent Using Spaces
Widths 2 Tab 2 Indent
Wrap lines
Área de Utilidades
View Controller - A controller that manages a view.
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.
Navigation Controller - A controller that manages navigation through a hierarchy of views.

Ready | Today at 16:56

MyPlayground

```swift
//: Playground - noun: a place where people can play

import Foundation

let N = 100.0
var r = 4.0
let margem = 1.0
var continuar = true
while continuar == true {
  r = (r + N/r)/2
  if ((r * r < N - margem) || (r * r > N + margem)) {
    continuar = true
  } else {
    continuar = false
  }
}
print (r)



// Várias perguntas
let p1 = "Qual a capital da Croácia?"
let respostaP1 = "Zagreb"

let p2 = "Em que ano Python 1.0 tornou-se disponível?"
let respostaP2 = "1994"

let p3 = "Qual o nome do segundo álbum do Vampire Weekend?"
let respostaP3 = "Contra"
```

```
100
4
1
true

(3 times)

(2 times)

false


"10.0227882130651\n"


"Qual a capital da Croácia?"
"Zagreb"

"Em que ano Python 1.0 tor...
"1994"

"Qual o nome do segundo...
"Contra"
```

```
10.0227882130651
```

Aproveitando, vamos até o Playground…

Implemente um programa que, dados três números inteiros guardados em três variáveis, diz qual o menor, qual o do meio e qual o maior

Implemente uma função `bubbleSort` em Swift

Implemente uma função `quickSort` em Swift

Crie uma variável contendo um array representando um grafo usando uma lista (array) de adjacências. Os rótulos dos nós do grafo devem ser números inteiros. Implemente uma função `dfs` que, dados um array com essa característica, o rótulo de um nó inicial e o rótulo do nó a ser encontrado, verifica se este último é alcançável a partir do nó inicial usando uma busca em profundidade.

Agora implemente uma função chamada `bfs` que funciona como dfs, mas realiza a busca em largura.

Modifique sua representação de grafo para que arestas passem a ter pesos.

Agora implemente uma função chamada shortestPath que usa o algoritmo de Dijkstra para calcular os menores caminhos entre todos os nós do grafo.