

FIAP GRADUAÇÃO

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Desenvolvimento Avançado iOS

PROF. GUSTAVO CALIXTO

Introdução ao Swift

INTRODUÇÃO AO SWIFT

- Aula de Hoje
 - Entendimento geral sobre a linguagem
 - Variáveis e estruturas básicas de programação.
- Git com códigos de exemplo
 - <https://github.com/gmcalixto/ios>

INTRODUÇÃO AO SWIFT

- Uma linguagem de programação desenvolvida pela Apple para a criação de apps.
- Lançada em 2014 na Apple WWDC.
 - *Apple Worldwide Developers Conference*
- É baseada nas linguagens Objective-C, Ruby, Python, C#, entre outras.
- Orientada à programação multiparadigma.



INTRODUÇÃO AO SWIFT

- Não é uma linguagem superconjunto de C
- Frameworks Cocoa e Cocoa Touch.
 - *Cocoa: Foundation e AppKit*
 - *Cocoa Touch: Foundation e UIKit*
- É grátis e em código aberto
- Licença Apache 2.0.

```
var people = ["Dave", "Brian", "Alex", "A  
let name = "Alex"  
if let index = find(people, name) {  
    println("\(name) is person \(index +  
        delegate?.didFindPersonWithName(name,  
    } else {  
        println("Unable to find \(name) in "+  
    }
```

INTRODUÇÃO AO SWIFT

- Licença Apache 2.0
 - Permite o livre uso, redistribuição e alteração.
 - Não há exigência de reciprocidade.
 - A liberação de código fonte é opcional.
 - Exigido o *copyright*.
 - Não obriga o *copyleft*.
 - Livre de royalties e patentes.



INTRODUÇÃO AO SWIFT

Comunidade Swift

- www.swift.org

The screenshot shows the homepage of Swift.org. On the left is a sidebar with links: ABOUT SWIFT, BLOG, DOWNLOAD, GETTING STARTED, DOCUMENTATION, SOURCE CODE, COMMUNITY, CONTRIBUTING, CONTINUOUS INTEGRATION, PROJECTS, and COMPILER AND STANDARD LIBRARY. The main content area has a title "Welcome to Swift.org" and a sub-section "Swift is now open source!". It discusses the excitement of Swift becoming open source and its impact on the language's growth and accessibility. Below this is a section titled "The Swift Team".

The screenshot shows a page from the iOS Developer Library titled "The Swift Programming Language (Swift 2.2)". The left sidebar includes links for Welcome to Swift, About Swift, A Swift Tour, Language Guide, Language Reference, and Revision History. The main content is titled "Control Flow" and explains how to use if and switch statements. It includes a code snippet:

```
1 let individualScores = [75, 43, 103, 87, 12]
2 var teamScore = 0
3
4 for score in individualScores {
5     if score > 50 {
6         teamScore += 3
7     } else {
8         teamScore += 1
9     }
10
11 print(teamScore)
```

A note at the bottom states: "In an if statement, the conditional must be a Boolean expression—this means that code such as if score { ... } is an error, not an implicit comparison to zero."

INTRODUÇÃO AO SWIFT



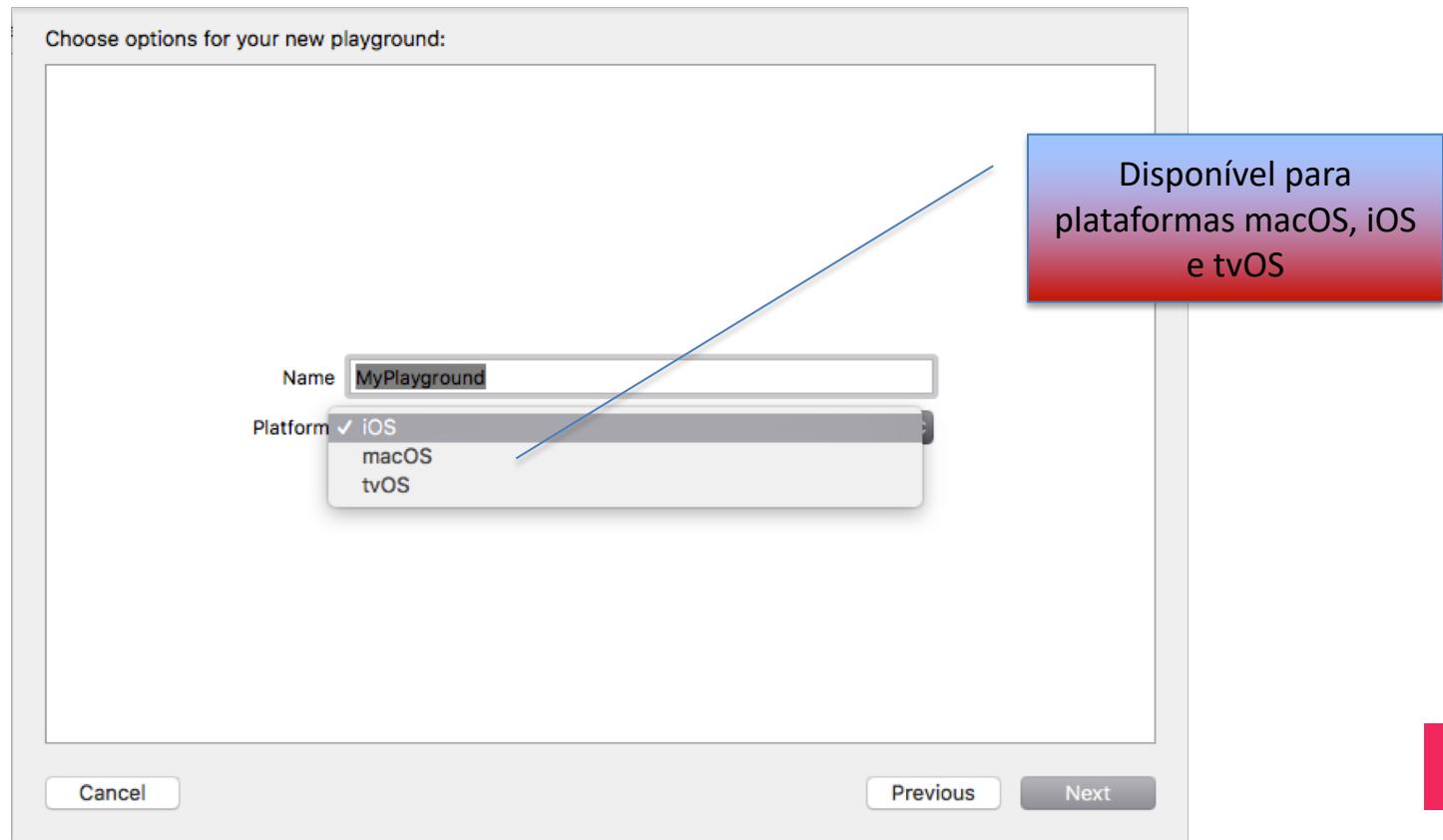
- Xcode Playgrounds
 - Ferramenta para treinamento de codificação.



INTRODUÇÃO AO SWIFT



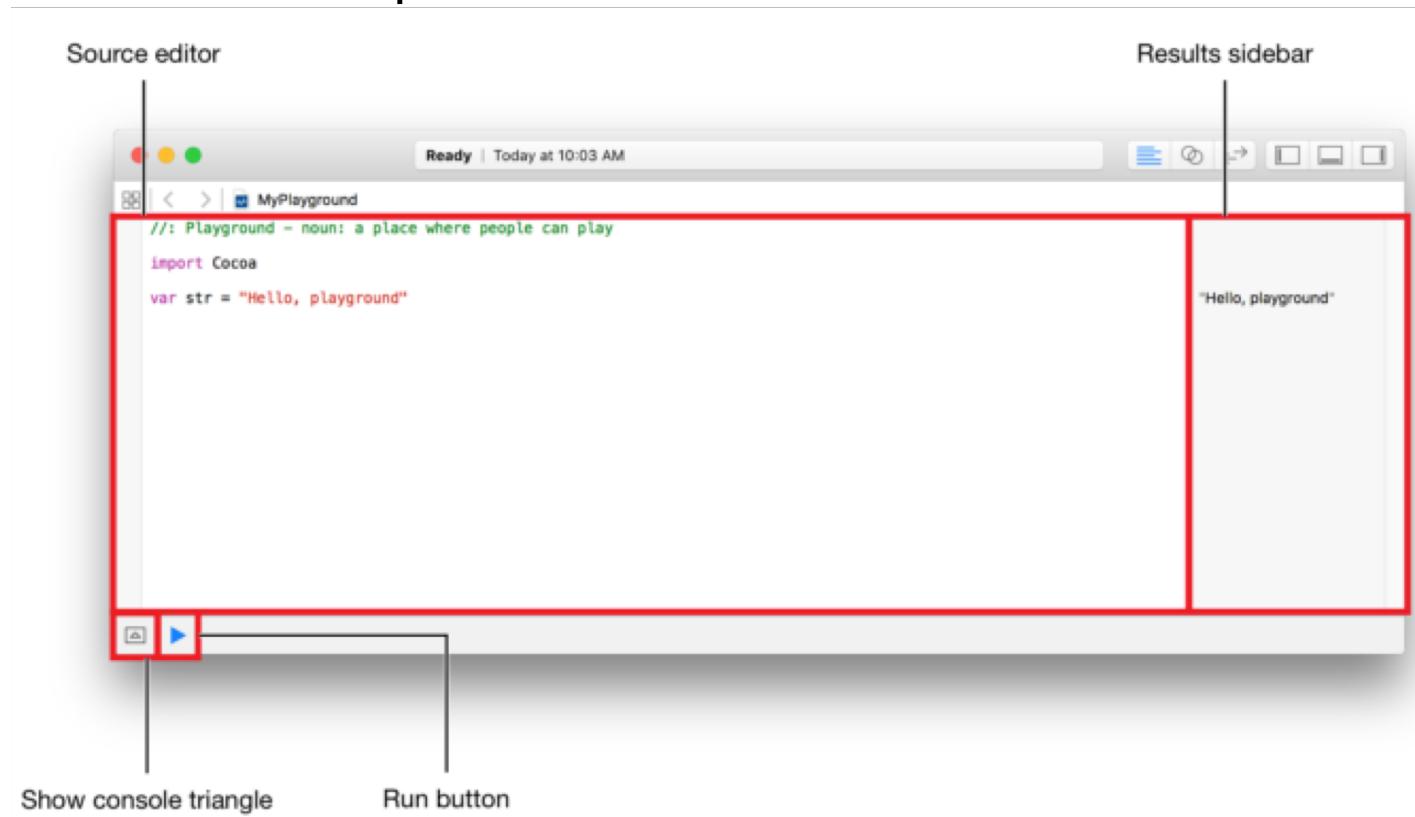
- Xcode Playgrounds
 - Selecionando plataforma para testes.



INTRODUÇÃO AO SWIFT



- XCode Playgrounds
 - Ferramenta para treinamento e teste de samples.



Fonte: Apple

INTRODUÇÃO AO SWIFT

- Interpretadores alternativos na Internet
 - Ferramenta para treinamento e teste de samples.
<https://iswift.org/playground>
<https://swift.sandbox.bluemix.net/#/repl>

INTRODUÇÃO AO SWIFT

Hello World

- Não é necessário o bloco main()

```
print("Hello, world!")
```

Não é necessário o uso
de ponto-e-vírgula ao
final da sentença

INTRODUÇÃO AO SWIFT

Variáveis simples

- var para variável
 - Usado para armazenamento direto de valor
- let para constantes
 - Usado para armazenar referências de objetos ou qualquer outro binding dinâmico
- tipo pode ser implícito na inicialização

```
1  var myVariable = 42  
2  myVariable = 50  
3  let myConstant = 42
```

INTRODUÇÃO AO SWIFT

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

//import Cocoa

var str = "Hello, playground"

print (str)

var minhaVariavelInteira = 50      Declaração implícita
var minhaVariavelFloat = 30.0
var minhaVariavelExplicitaInt: Int8 = 10
var minhaVariavelExplicitaFloat: Float = 234.57      Declaração explícita
```

```
print("Variavel com casting: " + String(minhaVariavelInteira))
print("Variavel Float com barra invertida: \(minhaVariavelFloat)")          Uso de typecasting

print("Variavel com casting: " + String(minhaVariavelExplicitaFloat))
print("Variavel Float com barra invertida: \(minhaVariavelExplicitaInt)")
```

INTRODUÇÃO AO SWIFT

Arrays e dicionários

```
1 var shoppingList = ["catfish", "water",
                     "tulips", "blue paint"]
2 shoppingList[1] = "bottle of water"
3
4 var occupations = [
5     "Malcolm": "Captain",
6     "Kaylee": "Mechanic",
7 ]
8 occupations["Jayne"] = "Public Relations"
```



Criação do vetor inicializado



Acessando elemento do vetor



Dicionário



Acessando elemento
do dicionário

INTRODUÇÃO AO SWIFT

Arrays e dicionários

```
var meuVetorNumerosInt = [10,20,30,40]
var meuVetorVazioExplicito = [String]()
```

Vetor inicializado com inteiros

```
meuVetorVazioExplicito.append("Maria")
meuVetorVazioExplicito.append("Joao")
meuVetorVazioExplicito.append("Mario")
meuVetorVazioExplicito.append("Manoel")
```

Binding de vetor dinâmico de Strings

Adiciona elementos à estrutura instanciada.

Label para impressão de coleções

```
print ("Número: \(meuVetorNumerosInt)")
print ("Número: \(meuVetorNumerosInt[0])")
```

Imprime todos os elementos

Imprime um elemento específico

```
print("Nomes:" + String(describing: meuVetorVazioExplicito))
print("Nome: \(meuVetorVazioExplicito[2])")
```

Casting de int para String

```
print("Elementos: \(meuVetorVazioExplicito.count)")
```

Conta elementos do vetor

```
print(meuVetorVazioExplicito.index(of: "Mario")!)
print(meuVetorVazioExplicito.index(of: "Joao")!)
```

Retorna o índice do elemento no vetor. O caractere "!" força o uso do dado mesmo que nulo.

```
meuVetorVazioExplicito = []
print("Vetor vazio:" + String(describing: meuVetorVazioExplicito))
```

INTRODUÇÃO AO SWIFT

Arrays e dicionários

```
//dicionarios

var meuDicionarioImplicito = [
    "Gustavo":"555-1010",
    "Diego":"555-1234",
    "Daniela":"555-4321"]

print("Dicionario Implicito: \(meuDicionarioImplicito["Gustavo"])")
print("Dicionario Implicito: \(meuDicionarioImplicito["Diego"])")

var meuDicionarioExplicito = [String:Float]()
meuDicionarioExplicito["cafe"] = 5.00
meuDicionarioExplicito["pao"] = 7.50
meuDicionarioExplicito["leite"] = 9.75

print("Preco do cafe: \(meuDicionarioExplicito["cafe"])")

meuDicionarioExplicito = [:]

print("Dicionario vazio: \(meuDicionarioExplicito)")
```

Estrutura que define par de dados

Par de dados explícitos

Recuperação de dados do dicionário

Atribuindo dicionário vazio

INTRODUÇÃO AO SWIFT

Estrutura de condição e laços
if-else

```
//estrutura de condição

var valor = 90

if valor > 50
{
    print("Maior")
}
else if (valor % 2) == 0
{
    print("É par")
}
else
{
    print("Ímpar")
}
```

Não há necessidade dos parênteses

INTRODUÇÃO AO SWIFT

Estrutura de condição e laços switch

```
//estrutura de escolha

var carro = "carro"

switch carro
{
    case "ford": print("Ford")
    case "vw": print("VW")
    case "fiat": print("FIAT")
    case "renault","citroen":
        print("Frances")
        print("Tem a peugeot")
    case let a where a.hasPrefix("ca"):
        print("Alguma coisa")
    default:
        print("Não encontrei")
        print("Tente outra vez")
}
```

Switch pode operar com quaisquer tipos de dados

Resultado lógico baseado em
resposta de métodos

INTRODUÇÃO AO SWIFT

Estrutura de condição e laços for-in

```
//laço for-in

let notas = [7.0, 8.5, 9.5, 10.0, 4.5, 6.0, 9.0]

for nota in notas{ → Percorre todos os elementos de uma estrutura

    if nota >= 6.0{

        print("Aprovado")
    }
    else{

        print("Reprovado")
    }
}
```

INTRODUÇÃO AO SWIFT

Estrutura de condição e laços for

Legado de linguagens similares
Entretanto, não recomendado no Swift

```
//laço for (deprecated)|  
  
for var count = 0; count < 10; count++ {  
    print(count);  
}
```

```
//laço for (recomendado)  
  
for count in 0 ..< 10 {  
    print(count);  
}
```

"count in x..< y" define intervalo de iterações do for
controlado na variável "count"

INTRODUÇÃO AO SWIFT

Estrutura de condição e laços while / repeat-while

```
//laço while
var n = 2
while n < 100 {
    n *= 2;
}
print (n);

//laço repeat-while
var m = 2
repeat {
    m *= 2;
} while m < 100
print (m)
```

INTRODUÇÃO AO SWIFT

Uso do Optional

- Informa que a variável pode ter valor nulo

```
//uso do Optional
var nomeOp:String? = nil
var nome:String = "Amador"

//se nomeOp for nulo, usa a variavel nome
print (String(nomeOp ?? nome))

nomeOp = "Agnaldo"
//com nomeOp não nulo
print (String(nomeOp ?? nome))
```

"?" indica que a variável pode ter valor nulo (nil)

Usa valor de redundância se variável de teste for nula

INTRODUÇÃO AO SWIFT

Exercícios de Fixação

- Desenvolva um algoritmo que imprima os 20 primeiros número de uma série de Fibonacci. Indicar se os números são primos, pares ou ímpares.
- Crie uma lista com o nome de 20 pessoas e suas respectivas idades. Depois, monte um laço no qual separe as pessoas com menos de 18 anos, mais de 60 anos, pessoas com nomes que terminam por “a” e pessoas que começem o nome por “e”.

INTRODUÇÃO AO SWIFT

- Simulando uma fatura mensal de um cartão de crédito, crie uma lista simulando 20 itens de despesas com os respectivos valores. O algoritmo deve calcular o valor total da fatura e também o valor do pagamento mínimo (10% do valor total da fatura).
- Crie um algoritmo que informa se uma variável Optional possui valor nulo. Caso o valor da variável seja nulo, imprimir a mensagem “Valor nulo”. Senão, deve ser impresso o conteúdo da variável.

Copyright © 2019 Prof. Gustavo Moreira Calixto

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).