

# Classes IOS – Swift

Parte 3 – Exceptions e Protocolos

X-Code

Prof. Agesandro Scarpioni

[agesandro@fiap.com](mailto:agesandro@fiap.com)



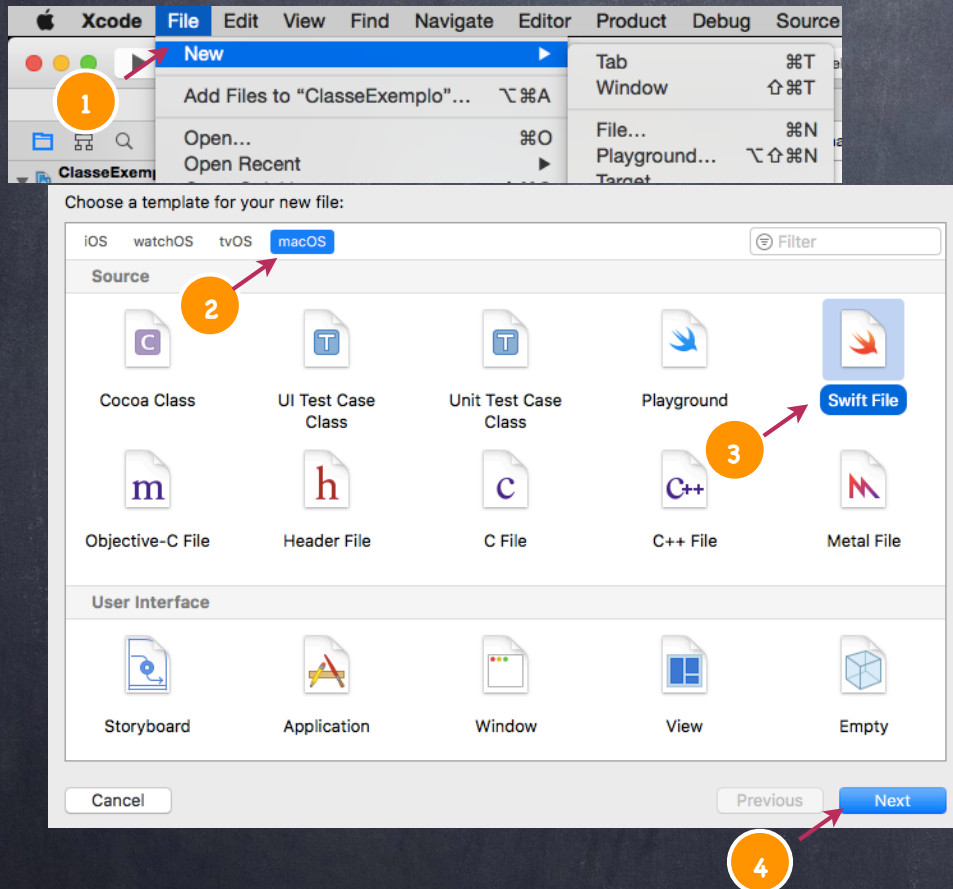
# Protocolos

- O protocolo funciona como as interfaces do Java ou do VB ou aos métodos virtuais do C++, é declarado com a notação protocol e uma classe pode implementar um ou mais protocolos, desde que quando tivermos mais que um protocolo, a declaração entre um protocolo e outro deve ser separado por vírgula, exemplo: protocolo1, protocolo2, protocolo3.

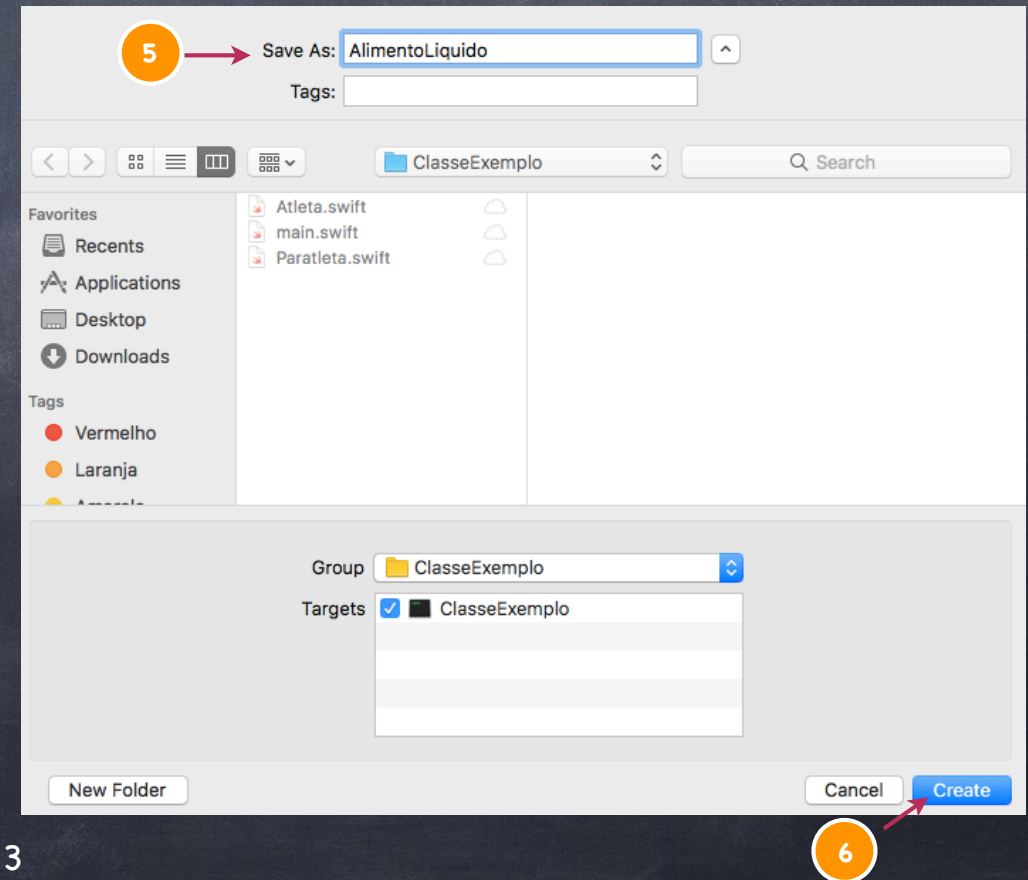


# Protocolos

- Crie 2 protocolos para alimentação do atleta, AlimentoLiquido, AlimentoSolido. Clique em File->New->File->macOS->Source->Swift File.



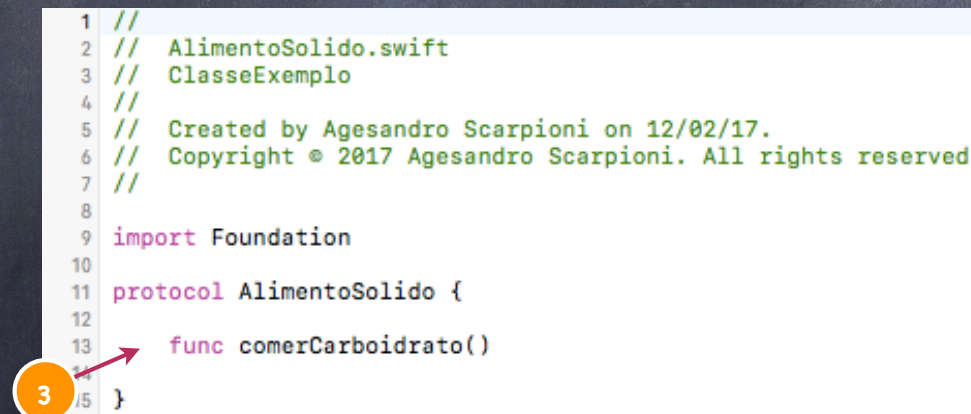
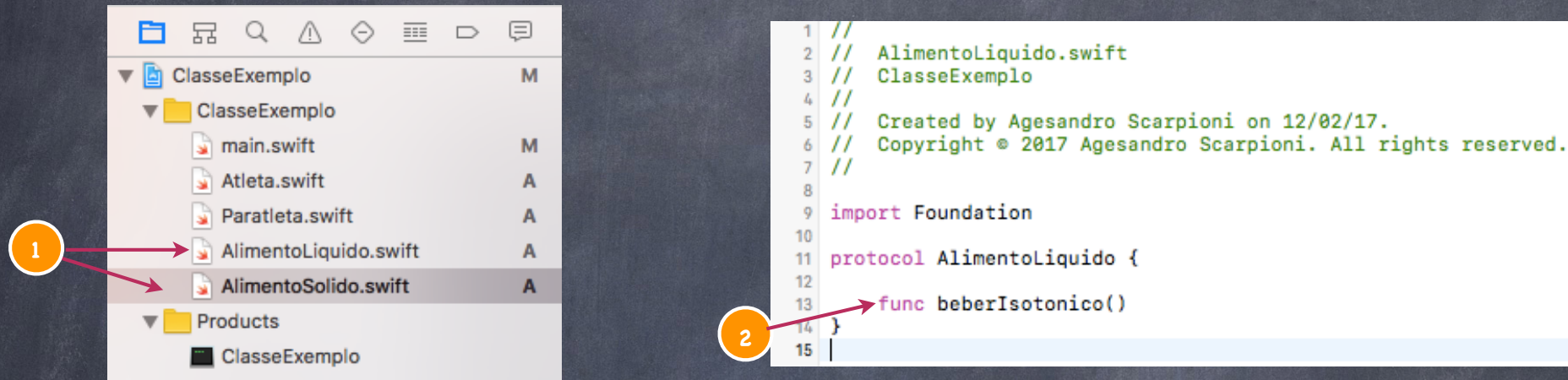
- Nomeie o primeiro como AlimentoLiquido e clique em Create, repita os passos para o AlimentoSolido.





# Protocolos

- Com os 2 protocolos criados como mostra na imagem 1, declare os métodos destacados nos pontos 2 e 3.



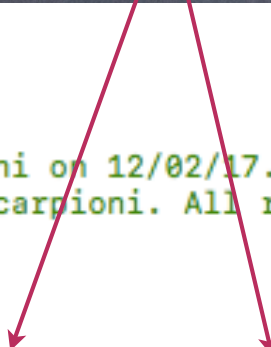
**Obs:** Também é possível ter o protocolo `AlimentoLiquidoSolido` que herdaria os métodos dos 2 protocolos como foi feito nas aulas de Objective-C.



# Protocolos

- Na classe Atleta.swift declare os dois protocolos de AlimentoLiquido e AlimentoSolido separados por vírgula.

```
1 //  
2 //  Atleta.swift  
3 //  ClasseExemplo  
4 //  
5 //  Created by Agesandro Scarpioni on 12/02/17.  
6 //  Copyright © 2017 Agesandro Scarpioni. All rights reserved.  
7 //  
8  
9 import Cocoa  
10  
11 class Atleta: NSObject, AlimentoLiquido, AlimentoSolido {  
12  
13     var nome:String  
14     var idade:Int  
15 }
```





# Protocolos

- Na classe `Atleta.swift` implemente os dois métodos dos dois protocolos `AlimentoLiquido` e `AlimentoSolido`.

```
41 func descricao() ->String{
42     return "O Atleta " + self.nome
43 }
44
45 //métodos de classe, no Obj-C se usa o símbolo (+) no início do nome do método
46 //O método de classe é como o método estático no Java
47 class func alerta()->String{
48     return "Se beber não dirija"
49 }
50
51 func beberIsotonico() {
52     print("Atleta \(self.nome) deve beber Gatorade")
53 }
54
55
56 func comerCarboidrato() {
57     print("Atleta \(self.nome) deve comer Batata Doce")
58 }
59
```



# Protocolos

- Na main.swift teste os protocolos chamando os dois métodos, um para cada objeto de atleta.

```
1 //
2 //  main.swift
3 //  ClasseExemplo
4 //
5 //  Created by Agesandro Scarpioni on 12/02/17.
6 //  Copyright © 2017 Agesandro Scarpioni. All rights reserved.
7 //
8
9 import Foundation
10
11 var a = Atleta()
12
13 a.nome = "Ana"
14 a.idade = 22
15
16 print("NOME: \(a.nome)")
17 print("IDADE: \(a.idade)")
18
19 print("Iron Man \(a.nome) \(a.idade) anos")
20
21 var a2 = Atleta(nome: "Carlos Gomes", idade: 25)
22 print("Iron Man \(a2.nome) \(a2.idade) anos")
23
24 //chamando uma Sub-rotina que exibe um texto
25 a2.calcularIMC(peso: 100, altura: 1.85)
26
27 //imprimindo os dados de uma função que retorna um float
28 print(a.calcularIMC_2(peso: 80, altura: 1.75))
29
30 let p = Paratleta()
31 p.nome = "Jonas"
32 p.idade = 30
33 p.modalidade = "natação"
34 print(p.descricao())
35
36 a.beberIsotonico()
37 a2.comerCarboidrato()
38
```

1

2



# Protocolos

- Veja que cada objeto (atleta) chamou um método de cada protocolo, observe o resultado.

```
35  
36 a.beberIsotonico()  
37 a2.comerCarboidrato()  
38
```

```
NOME: Ana  
IDADE: 22  
Iron Man Ana 22 anos  
Iron Man Carlos Gomes 25 anos  
O IMC de Carlos Gomes é 29.2184  
26.1224  
O Atleta Jonas da natação é paratleta  
Atleta Ana deve beber Gatorade  
Atleta Carlos Gomes deve comer Batata Doce  
Program ended with exit code: 0
```



# Exceptions

- A forma de criar uma exception em Swift é muito semelhante a forma de criação em outras linguagens.



# Exceptions

- Ainda na classe `Atleta.swift` faça o lançamento da exception dentro do método `CalcularImc`, use no final da função a palavra reservada `throws`, em seguida monte o `if` para não permitir altura maior que 2 metros.

```
26
27 //Sub-rotina (não tem retorno) funciona como o Void em várias linguagens.
28 func calcularIMC(peso:Float, altura:Float) throws{
29     if altura > 2 {
30         let msg = "Altura não pode ser maior que 2 metros"
31         throw NSError(domain: msg, code: 0, userInfo: nil)
32     }
33     let imc = peso / (altura * altura)
34     print("O IMC de \(self.nome) é \(imc)")
35 }
```



# Exceptions

- Na classe main.m use o Do/Try/Catch para capturar o erro, vamos chamar um método com uma altura não permitida, por exemplo: 2.20m.

```
23  
24 //chamando uma Sub-rotina que exibe um texto  
25 do {  
26     try a2.calcularIMC(peso: 100, altura: 2.20)  
27 } catch let error as NSError {  
28     print("Error: \(error.domain)")  
29 }
```



# Exceptions

- Execute e veja o resultado.

```
NOME: Ana
IDADE: 22
Iron Man Ana 22 anos
Iron Man Carlos Gomes 25 anos
Error: Altura não pode ser maior que 2 metros
26.1224
O Atleta Jonas da natação é paratleta
Atleta Ana deve beber Gatorade
Atleta Carlos Gomes deve comer Batata Doce
Program ended with exit code: 0
```



# Classes + POO

- Continuaremos com exercícios práticos com MVC no próximo conjunto de slides.