

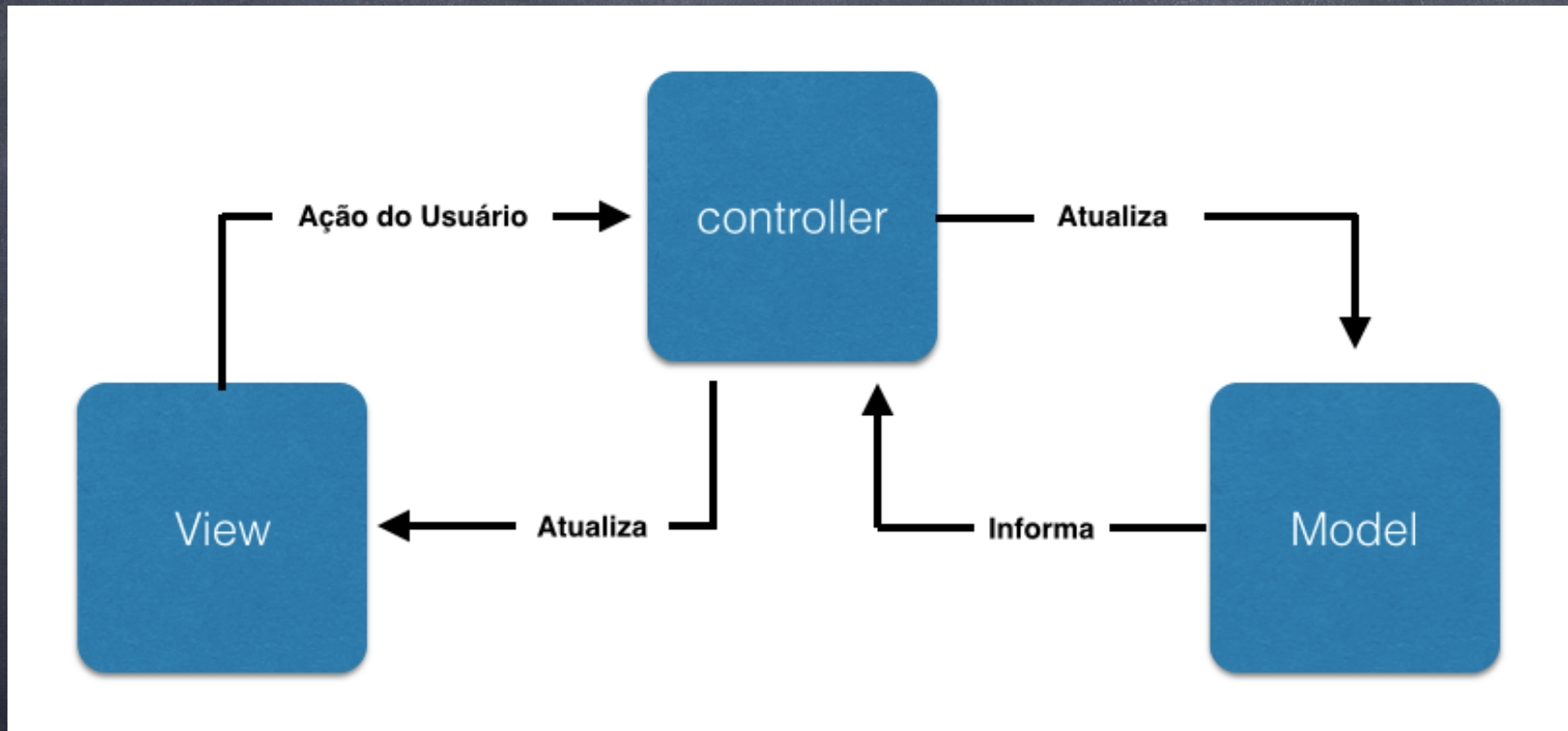
Classes IOS – MVC

X-Code – Swift
Prof. Agesandro Scarpioni
agesandro@fiap.com.br

MVC – Model View Controller

- MVC é um padrão de projeto, padrão de projeto ou Design Patterns pensando em uma forma simples de explicar, são soluções para problemas que alguém um dia teve e resolveu aplicando um modelo que foi documentado e que podemos adaptar em nossa solução.
- O MVC é uma forma de estruturar o projeto/aplicação de forma que a interface de interação (View) seja separada do controle da informação (Models), essa separação é intermediada por outra camada controladora (Controllers).

MVC – Model View Controller



Criar o Projeto

- Em um projeto do tipo Single View Application crie um projeto chamado Exemplo_MVC_Swift.

Choose a template for your new project:

iOS watchOS tvOS macOS C...

Application

1
Single View Application

Game

Sticker Pack Application

iMessage Application

Framework & Library

Cocoa Touch Framework

Cocoa Touch Static Library

Choose options for your new project:

Product Name: Exemplo_MVC_Swift

Team: Add account...

Organization Name: Agesandro Scarpioni

Organization Identifier: com.scarpioni

Bundle Identifier: com.scarpioni.Exemplo-MVC-Swift

Language: Swift

Devices: iPhone

☐ Use Core Data

☐ Include Unit Tests

☐ Include UI Tests

Cancel Previous Next

Criar o Projeto

- Crie no Storyboard a seguinte View com label's, Text's e um Button.

The image shows a storyboard view of a 'View Controller'. It contains a form with four labels and text input fields, and a button. The labels are 'Atleta:', 'Peso:', 'Altura:', and 'IMC:'. The input fields are empty. The button is labeled 'Calcular' in blue text. A grey arrow points to the left side of the view, indicating its position in the storyboard.

View Controller

Atleta:

Peso:

Altura:

IMC:

Calcular

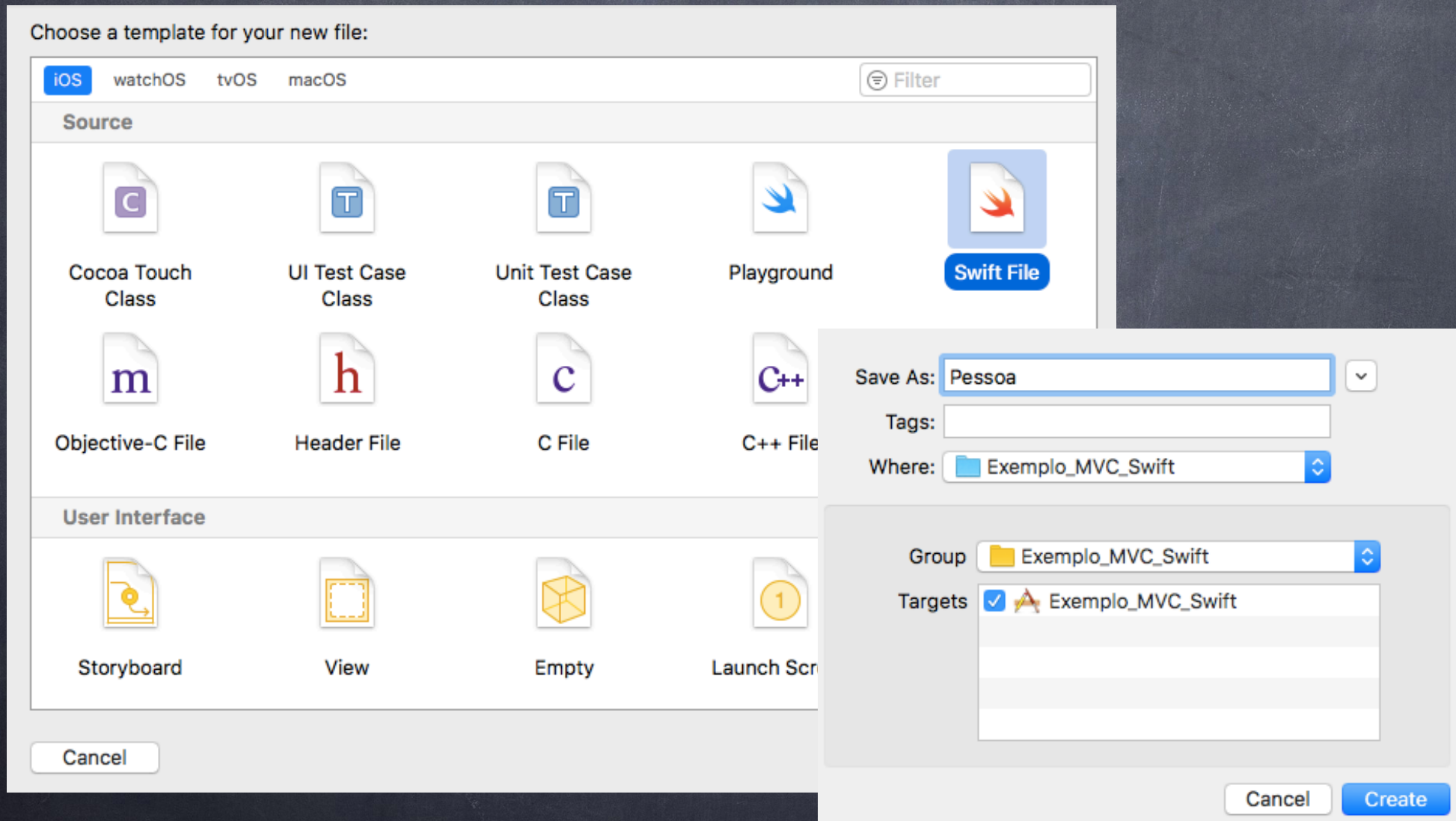
Outlet e Action

- No arquivo ViewController.swift, foi declarado como outlets todas as caixas de texto, além disso, foi criado um IBAction para o botão Calcular, ele irá disparar via TouchUpInside o método que está na model que nesse caso é a classe Pessoa.

```
1 //  
2 // ViewController.swift  
3 // Exemplo_MVC_Swift  
4 //  
5 // Created by Agesandro Scarpioni on 18/02/17.  
6 // Copyright © 2017 Agesandro Scarpioni. All rights reserved.  
7 //  
8  
9 import UIKit  
10  
11 class ViewController: UIViewController {  
12  
13  
14     @IBOutlet weak var txtNome: UITextField!  
15     @IBOutlet weak var txtPeso: UITextField!  
16     @IBOutlet weak var txtAltura: UITextField!  
17     @IBOutlet weak var txtImc: UITextField!  
18  
19  
20     @IBAction func calcular(_ sender: Any) {  
21  
22     }  
23  
24     override func viewDidLoad() {  
25         super.viewDidLoad()  
26         // Do any additional setup after loading the view,  
27         // typically from a nib.  
28     }  
29  
30     override func didReceiveMemoryWarning() {  
31         super.didReceiveMemoryWarning()  
32         // Dispose of any resources that can be recreated.  
33     }  
34 }  
35  
36
```


Criar a Model

- Crie uma classe (Command + N) em Swift chamada Pessoa.



Classe Pessoa

- Escreva `class Pessoa { }` e prepare-se para implementar a classe.

```
1 //  
2 // Pessoa.swift  
3 // Exemplo_MVC_Swift  
4 //  
5 // Created by Agesandro Scarpioni on 18/02/17.  
6 // Copyright © 2017 Agesandro Scarpioni. All rights  
  reserved.  
7 //  
8  
9 import Foundation  
10  
11 class Pessoa{  
12  
13  
14 }  
15
```


Métodos

- Nessa classe foi criado os atributos e métodos de Pessoa. Criei dois métodos que fazem a mesma coisa, porém, um método é do tipo "Void", ou seja, não tem retorno, e outro do tipo Float que retorna o valor calculado para quem o chamar.

```
1 //
2 // Pessoa.swift
3 // Exemplo_MVC_Swift
4 //
5 // Created by Agesandro Scarpioni on 18/02/17.
6 // Copyright © 2017 Agesandro Scarpioni. All rights
  reserved.
7 //
8
9 import Foundation
10
11 class Pessoa{
12     var nome:String!
13     var peso:Float!
14     var altura:Float!
15     var imc:Float!
16
17     //um exemplo sem parâmetro do tipo void
18     func calcularIMC(){
19         imc=peso/(altura*altura)
20     }
21     //um exemplo sem parâmetro com retorno de um float
22     func calcularIMC_2() -> Float{
23         return peso/(altura*altura)
24     }
25 }
26
```

Obs: Em Swift uma classe que tem retorno e uma classe que não tem retorno, ambas são declaradas com func, as que tem return são as funções e as que não possuem return são como o void do Java.

Construtores

- Para ilustrar foi criado na classe um construtor padrão "init()" e um construtor com 4 parâmetros

```
1 //
2 // Pessoa.swift
3 // Exemplo_MVC_Swift
4 //
5 // Created by Agesandro Scarpioni on 18/02/17.
6 // Copyright © 2017 Agesandro Scarpioni. All rights reserved.
7 //
8
9 import Foundation
10
11 class Pessoa{
12     var nome:String!
13     var peso:Float!
14     var altura:Float!
15     var imc:Float!
16
17     //um exemplo sem parâmetro do tipo void
18     func calcularIMC(){
19         imc=peso/(altura*altura)
20     }
21     //um exemplo sem parâmetro com retorno de um float
22     func calcularIMC_2() -> Float{
23         return peso/(altura*altura)
24     }
25     //criando um construtor padrão
26     init(){
27
28     }
29     //criando um construtor com 4 parâmetros
30     init(nome:String, peso:Float, altura:Float, imc:Float){
31         self.nome = nome
32         self.peso = peso
33         self.altura = altura
34         self.imc = imc
35     }
36 }
37
```


Exemplificando

- Abaixo faço a criação do objeto "p", ou seja, uma instância da classe Pessoa (linha 20). Da linha 21 até a 24 passo os dados da tela (view) para os atributos do objeto "p" da classe Pessoa (model). Na linha 24 foi chamado o método do tipo void, já na linha 25 é exibido o conteúdo de imc do objeto "p" na caixa txtImc da View.

```
1 //
2 // ViewController.swift
3 // Exemplo_MVC_Swift
4 //
5 // Created by Agesandro Scarpioni on 18/02/17.
6 // Copyright © 2017 Agesandro Scarpioni. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13
14     @IBOutlet weak var txtNome: UITextField!
15     @IBOutlet weak var txtPeso: UITextField!
16     @IBOutlet weak var txtAltura: UITextField!
17     @IBOutlet weak var txtImc: UITextField!
18
19     @IBAction func calcular(_ sender: Any) {
20         let p:Pessoa = Pessoa()
21         p.nome = txtNome.text
22         p.peso = Float(txtPeso.text!)
23         p.altura = Float(txtAltura.text!)
24         p.calcularIMC()
25         txtImc.text = "\(p.imc!)"
26     }
27
28     override func viewDidLoad() {
29         super.viewDidLoad()
30         // Do any additional setup after loading the view,
31         // typically from a nib.
32     }
33
34     override func didReceiveMemoryWarning() {
35         super.didReceiveMemoryWarning()
36         // Dispose of any resources that can be recreated.
37     }
38 }
```


- Abaixo a função calcular nas duas formas void (24 e 25) e a que retorna o Float(28), escolha uma das formas.

```
24      p.calcularIMC()  
25      txtImc.text = "\n(p.imc!)"  
26  
27      //chamando o IMC pela função que retorna o float  
28      txtImc.text = "\n(p.calcularIMC_2())"
```

- Para controlar a quantidade de casas decimais use:

```
txtImc.text = String(format: "%.2f", p.imc)
```


Exemplificando

- Exemplo completo do código do IBAction (botão Calcular).

```
19  @IBAction func calcular(_ sender: Any) {  
20      let p:Pessoa = Pessoa()  
21      p.nome = txtNome.text  
22      p.peso = Float(txtPeso.text!)  
23      p.altura = Float(txtAltura.text!)  
24      p.calcularIMC()  
25      txtImc.text = "\(p.imc!)"  
26  
27      //chamando o IMC pela função que retorna o float  
28      txtImc.text = "\(p.calcularIMC_2())"  
29  }
```


Run

- Nesse ponto, foi digitado os dados na View que foram passados para a Model via Controller e a mesma recebeu o cálculo da model e atualizou o IMC na View..

Carrier 3:36 PM

Atleta: Agesandro

Peso: 93

Altura: 1.85

IMC: 27.17

Calcular

Prática

Criação de um programa para testarmos todos os conceitos deste tópico.

- Crie um projeto para Restaurante com uma Classe Cliente que possua seus atributos e métodos, um exemplo de método que você possa implementar é PagarConta que retorna o cálculo do troco quando o valor for recebido em dinheiro, esse método precisa receber o valor da conta e o valor pago.
- Um método ReservarMesa que tenha como parâmetro inteiro de entrada a hora cheia e retorne a palavra "almoço" ou "jantar" dependendo do horário.
- Exemplo: Entre 11 e 15 exiba "almoço", entre 18 e 23 exiba "jantar", caso contrário exiba "horário indisponível para reserva."