

Classes IOS – MVC

X-Code – Obj-C

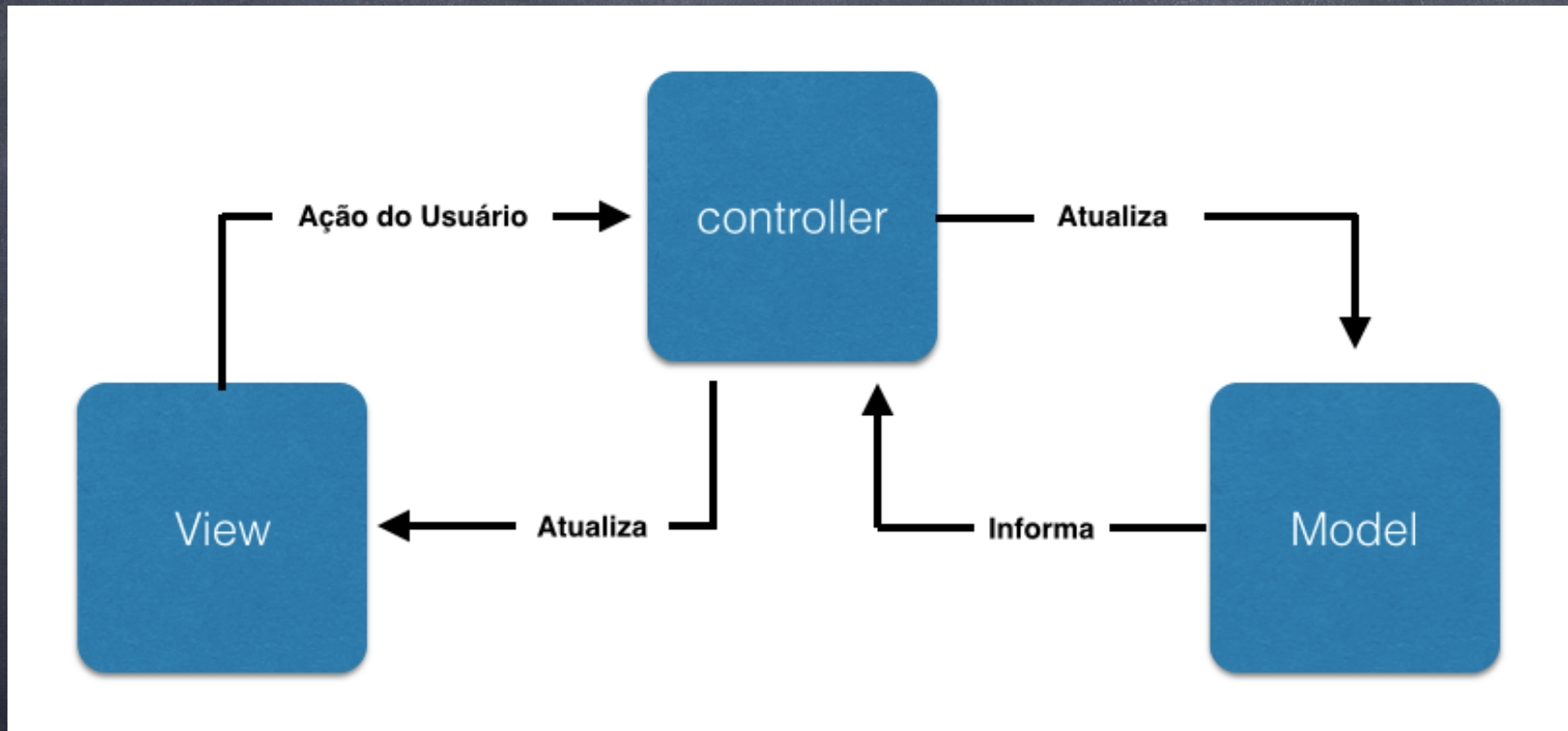
Prof. Agesandro Scarpioni

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 Design Patterns, 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



Exemplificando

- Em um projeto chamado Exemplo_ObjC, foi criado 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 four text labels with corresponding text input fields: 'Atleta:', 'Peso:', 'Altura:', and 'IMC:'. Below these fields is a blue button labeled 'Calcular'. A grey arrow points from the left towards the 'Atleta:' label and its input field.

Outlet e Action

- No arquivo .h (controller), foi declarada como outlets todas as caixas de texto utilizando @property, ou seja, eles serão nossos get/sets para a tela, além disso, foi criado um IBAction para o botão Calcular, ele irá disparar via TouchUpInside o método que está na model que é a classe Pessoa.

```
1 //  
2 // ViewController.h  
3 // Exemplo_MVC  
4 //  
5 // Created by Usuário Convidado on 22/04/15.  
6 // Copyright (c) 2015 Usuário Convidado. All rights reserved.  
7 //  
8  
9 #import <UIKit/UIKit.h>  
10  
11 @interface ViewController : UIViewController  
12  
13  
14 @property (weak, nonatomic) IBOutlet UITextField *txtNome;  
15  
16 @property (weak, nonatomic) IBOutlet UITextField *txtPeso;  
17  
18 @property (weak, nonatomic) IBOutlet UITextField *txtAltura;  
19  
20 @property (weak, nonatomic) IBOutlet UITextField *txtIMC;  
21  
22  
23 - (IBAction)calcular:(id)sender;  
24  
25 @end  
26  
27
```


Criar a Model

- Foi criada uma classe (Command + N) em Objective C chamada Pessoa, do tipo Cocoa Touch Class filha de NSObject. Nessa classe no arquivo .h 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.h
3 // Exemplo_MVC
4 //
5 // Created by Usuário Convidado on 22/04/15.
6 // Copyright (c) 2015 Usuário Convidado. All rights reserved.
7 //
8
9 #import <Foundation/Foundation.h>
10
11 @interface Pessoa : NSObject
12
13 @property (nonatomic, retain) NSString* nome;
14 @property (nonatomic) float peso;
15 @property (nonatomic) float altura;
16 @property (nonatomic) float imc;
17
18 //um exemplo sem parâmetro do tipo void
19 -(void) calcularIMC;
20 //um outro exemplo sem parâmetro porém retorna um float
21 -(float) calcularIMC_2;
22
23 @end
24
```


Classe Pessoa

- No arquivo .m da classe Pessoa (Model), foi implementado os dois métodos que fazem o mesmo cálculo, porém, no método "CalcularIMC_2" dei o return de várias maneiras possíveis, utilizei várias notações: objeto.propriedade, [objeto propriedade] e _propriedade, também fiz o cálculo de forma simples e utilizando a função pow para calcular potência.

```
1 //  
2 // Pessoa.m  
3 // Exemplo_MVC  
4 //  
5 // Created by Usuário Convidado on 22/04/15.  
6 // Copyright (c) 2015 Usuário Convidado. All rights reserved.  
7 //  
8  
9 #import "Pessoa.h"  
10  
11 @implementation Pessoa  
12  
13 -(float) calcularIMC_2{  
14  
15     return self.peso / (self.altura*self.altura);  
16  
17     //ou  
18  
19     //return _peso/(_altura*_altura);  
20  
21     // ou  
22  
23     //return [self peso] / ([self altura]* [self altura]);  
24  
25     //caso queira usar pow  
26  
27     //return self.peso / pow(self.altura,2);  
28  
29 }  
30  
31 -(void) calcularIMC{  
32     //vou fazer apenas de uma forma  
33     self.imc = self.peso / pow(self.altura,2);  
34 }  
35  
36 @end  
37
```


Métodos

- Na Controller.m foi dado um import da classe Pessoa.h (linha 10), veja no próximo slide a implementação (linha 27).

```
1 //  
2 // ViewController.m  
3 // Exemplo_MVC  
4 //  
5 // Created by Usuário Convidado on 22/04/15.  
6 // Copyright (c) 2015 Usuário Convidado. All rights reserved.  
7 //  
8  
9 #import "ViewController.h"  
10 #import "Pessoa.h"  
11 @interface ViewController ()  
12  
13 @end  
14  
15 @implementation ViewController  
16  
17 - (void)viewDidLoad {  
18     [super viewDidLoad];  
19     // Do any additional setup after loading the view, typically from a nib.  
20 }  
21  
22 - (void)didReceiveMemoryWarning {  
23     [super didReceiveMemoryWarning];  
24     // Dispose of any resources that can be recreated.  
25 }  
26
```


Exemplificando

- Abaixo faço a criação do objeto "p", ou seja, uma instância da classe Pessoa (linha 29). Até a linha 42 passo os dados da tela (view) para o objeto "p" da classe Pessoa (model), para isso utilizei diversas notações possíveis, da mesma forma que fiz na Classe Pessoa.
- Dica: escolha uma notação e a utilize em seus projetos, claro que ficaria inviável você usar uma notação diferente a cada linha.

```
27 - (IBAction)calcular:(id)sender {
28
29     Pessoa *p = [[Pessoa alloc] init];
30
31     //passando os dados da tela usando notação ponto
32     p.peso = self.txtPeso.text.floatValue;
33
34     //passando os dados da tela usando notação espaço [ ]
35     [p setPeso:[self txtPeso]text]floatValue];
36
37     //passando os dados de uma terceira forma
38     p.peso = _txtPeso.text.floatValue;
39
40     //daqui pra baixo da forma que eu gosto
41     p.altura = self.txtAltura.text.floatValue;
42     p.nome = self.txtNome.text;
43 }
```


Exemplificando

- Abaixo chamo a função calcular (void) na linha 44 e exibo o resultado linha 45. Já na linha 49 eu chamo a função que retorna o float, na linha 52 utilizei outra notação para chamar o método que retorna o float.

Dica: Escolha uma das formas.

```
43  
44     [p calcularIMC]; //lembre-se que este é o void  
45     self.txtIMC.text = [NSString stringWithFormat:@"%f",p.imc];  
46  
47     //forma para quem fez o método retornando o float  
48     //essa forma seria do jeito que eu AGE faria  
49     self.txtIMC.text = [NSString stringWithFormat:@"%0.2f",[p calcularIMC_2]];  
50  
51     //forma para atribuir para o txt usando notação [ ]  
52     [[self txtIMC]setText:[NSString stringWithFormat:@"%0.2f",[p calcularIMC_2]]];  
53  
54 }  
55 @end  
56
```


Exemplificando

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

```
26
27 - (IBAction)calcular:(id)sender {
28
29     Pessoa *p = [[Pessoa alloc] init];
30
31     //passando os dados da tela usando notação ponto
32     p.peso = self.txtPeso.text.floatValue;
33
34     //passando os dados da tela usando notação espaço [ ]
35     [p setPeso:[self txtPeso]text]floatValue];
36
37     //passando os dados de uma terceira forma
38     p.peso = _txtPeso.text.floatValue;
39
40     //daqui pra baixo da forma que eu gosto
41     p.altura = self.txtAltura.text.floatValue;
42     p.nome = self.txtNome.text;
43
44     [p calcularIMC]; //lembre-se que este é o void
45     self.txtIMC.text = [NSString stringWithFormat:@"%f",p.imc];
46
47     //forma para quem fez o método retornando o float
48     //essa forma seria do jeito que eu AGE faria
49     self.txtIMC.text = [NSString stringWithFormat:@"%0.2f",[p calcularIMC_2]];
50
51     //forma para atribuir para o txt usando notação [ ]
52     [[self txtIMC]setText:[NSString stringWithFormat:@"%0.2f",[p calcularIMC_2]]];
53
54 }
55 @end
56
```


Run

- Nesse ponto, digitei os dados na View que foram passados para a Model via Controller que atualizou o IMC.

iOS Simulator - iPhone 6 - iPhone 6 / iOS 8.3...

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."