

Rest e Json no iOS

Insert / Chave

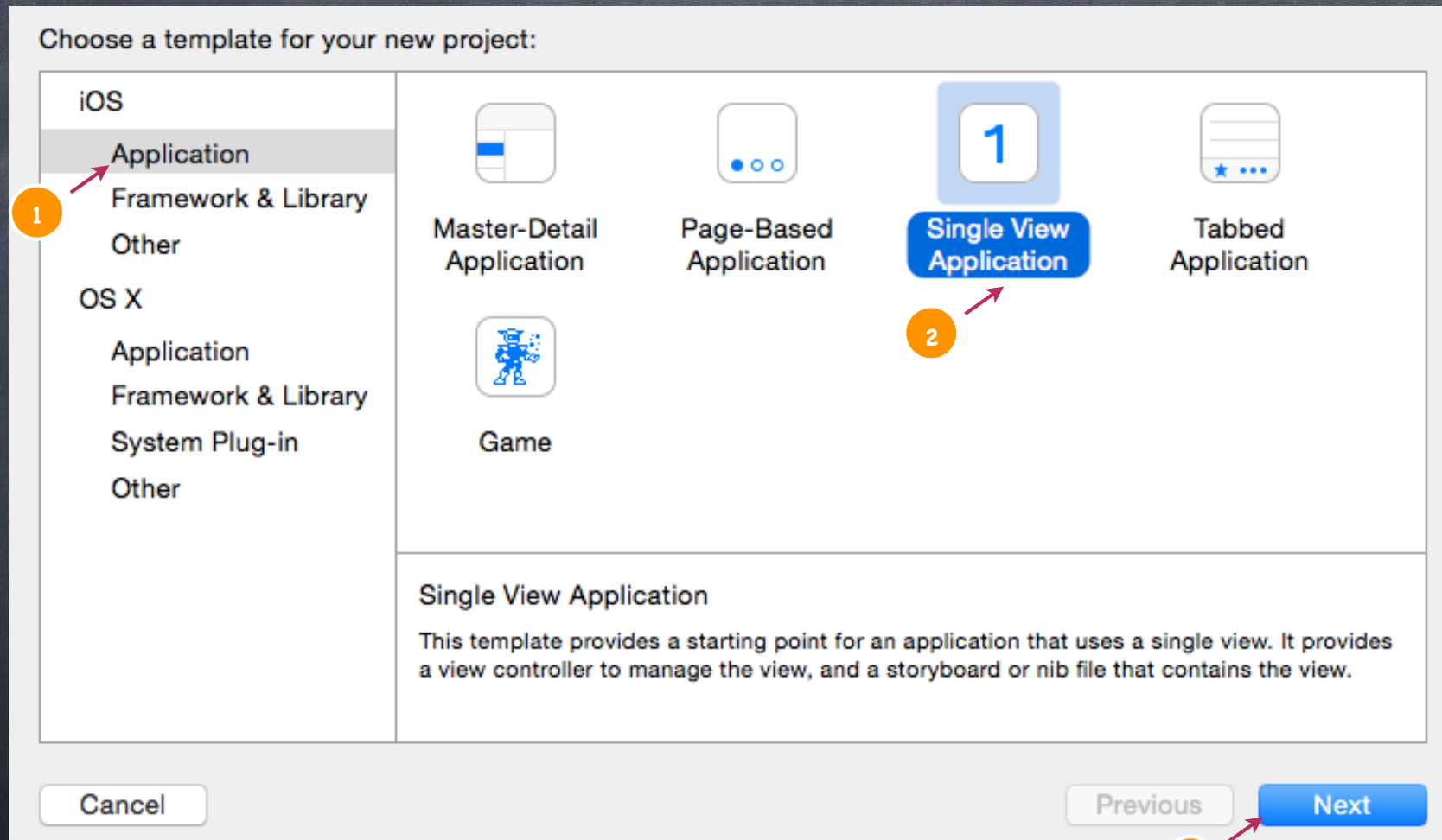
X-Code com Objective-C
Prof. Agesandro Scarpioni

App com label's, Text's e button

- Vamos criar um aplicativo para fazermos um request, e o parser do Json em um dicionário e gravar os dados digitados na tela.
- Em um segundo momento vamos fazer um request para exibir os dados em uma tabela.
- Note que é necessário seu RM tanto para gravar quanto para consultar, dessa forma exibiremos apenas os parques que você cadastrou.

Iniciando o Projeto

- Clique em File -> New Project -> iOS -> Application. -> Single View Application.



Os dados do Projeto

- Preencha com os dados abaixo, lembre-se que o Organization Identifier é como se fosse o pacote no Java ou o namespace do VB. Em Devices selecione iPhone.

Choose options for your new project:

Product Name:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:

Devices:

☒ Use Core Data

A Interface

- Desenhe a tela abaixo com 4 label's, 1 button e 4 Text's.

Local

Estado

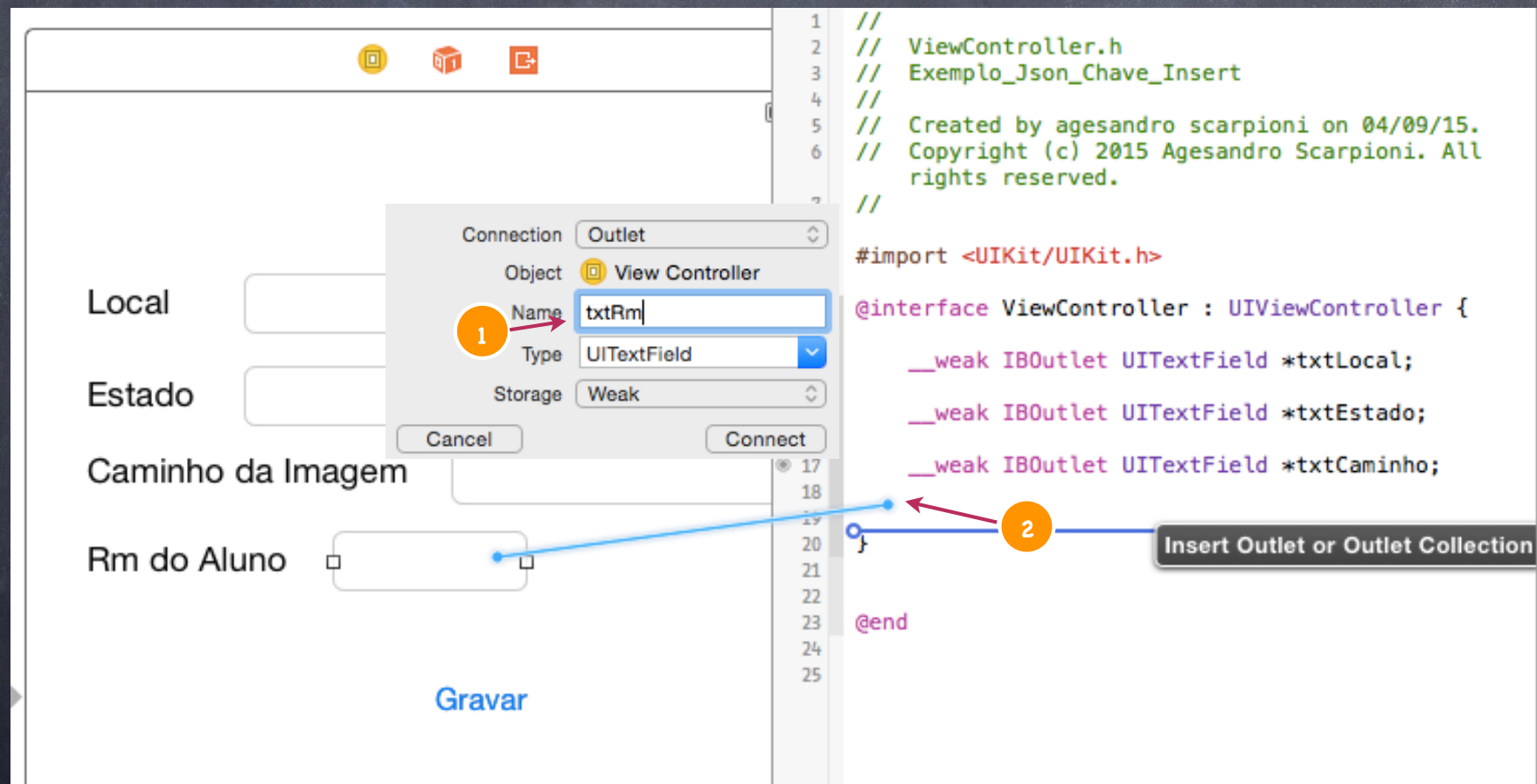
Caminho da Imagem

Rm do Aluno

Gravar

Definindo os IBOutlet's

- Abra duas telas simultâneas e vamos criar no .h os Outlet's dos Text's. Atenção coloque as chaves { } e com botão direito sobre o objeto (1), arraste dentro da área das chaves (2), repita o processo para todas as caixas de texto.



Definindo o IBAction

- Clique com o botão direito sobre o Gravar(1), arraste até a área fora das chaves { }, Nomeie o IBAction como gravar(2) e altere o Connection para Action(3).


The screenshot illustrates the steps to define an IBAction in Xcode. On the left, the Interface Builder (IB) shows a form with fields for 'Local', 'Estado', 'Caminho da Imagem', and 'Rm do Aluno'. A button labeled 'Gravar' is highlighted with a red circle and the number 1. A right-click on this button opens a 'Connect' dialog box. In this dialog, the 'Connection' is set to 'Action' (3), the 'Object' is 'View Controller' (2), and the 'Name' is 'gravar'. The 'Event' is 'Touch Up Inside' and the 'Arguments' are 'Sender'. A blue arrow points from the 'Connect' button in the dialog to the code editor. The code editor shows the ViewController.h file with the @interface block. A tooltip 'Insert Outlet, Action, or Outlet Collection' is visible near the @end line.

```
1 //  
2 // ViewController.h  
3 // Exemplo_Json_Chave_Insert  
4 //  
5 // Created by agesandro scarpioni on 04/09/15.  
6 // Copyright (c) 2015 Agesandro Scarpioni. All  
7 // rights reserved.  
8 //  
9 #import <UIKit/UIKit.h>  
10  
11 @interface ViewController : UIViewController {  
12     __weak IBOutlet UITextField *txtLocal;  
13     __weak IBOutlet UITextField *txtEstado;  
14     __weak IBOutlet UITextField *txtCaminho;  
15     __weak IBOutlet UITextField *txtRm;  
16 }  
17  
18 @end
```


Definindo os IBAction's

- Note que ao fazer a ligação do IBAction além da declaração do método no arquivo .h, também a área da implementação (1) aparece no arquivo .m.

```
1 //
2 // ViewController.m
3 // Exemplo_Json_Chave_Insert
4 //
5 // Created by agesandro scarpioni on 04/09/15.
6 // Copyright (c) 2015 Agesandro Scarpioni. All
7 // rights reserved.
8 //
9 #import "ViewController.h"
10
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
20     // view, typically from a nib.
21 }
22
23 - (void)didReceiveMemoryWarning {
24     [super didReceiveMemoryWarning];
25     // Dispose of any resources that can be
26     // recreated.
27 }
28
29 - (IBAction)gravar:(id)sender {
30 }
31 @end
```



Implementando o Gravar

- Nas linhas abaixo estamos fazendo o request e o parser do Json para um NSDictionary, a url abaixo é de um Web Service hospedado em meu servidor.

```
20
21
22
23
24
25
26
27 - (IBAction)gravar:(id)sender {
28
29     NSString *stringUrl = @"http://scarpioni.com/webservices/inserir.php";
30
31     //transformando a string em uma url
32     NSURL *url = [NSURL URLWithString:stringUrl];
33
34     //Criando uma requisição com a url informada
35     NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:url];
36
37     //Definindo o método como Post
38     [request setHTTPMethod:@"POST"];
39
40     //setando o tipo de conteúdo com json
41     [request setValue:@"application/json" forHTTPHeaderField:@"Content-Type"];
42
43     //setando uma chave para acesso aos dados
44     [request setValue:@"1234567890" forHTTPHeaderField:@"chave-api"];
45
46
47     NSMutableDictionary *body = [[NSMutableDictionary alloc] init];
48     //apesar de não termos a caixa de texto cidade somos obrigados a passar
49     //todas as chaves, por isso repasso a caixa de estado duas vezes
50     //não é possível fazer o insert faltando um campo, faça um teste sem
51     //a chave cidade e veja o erro que aparece, o webservice precisa de todos os campos.
52     [body setValue:txtLocal.text forKey:@"nome"];
53     [body setValue:txtEstado.text forKey:@"cidade"];
54     [body setValue:txtEstado.text forKey:@"estado"];
55     [body setValue:txtCaminho.text forKey:@"urlfoto"];
56     [body setValue:txtRm.text forKey:@"rm"];
57
58     //Serializa o Dictionary para Data
```


Implementando o Gravar

- Continuação.
- Execute esse trecho e verifique no Console os dados que subiram e os dados que retornaram do NSLog - linhas 67 e 82.

```
57
58 //Serializa o Dictionary para Data
59 NSError *error;
60 NSData *jsonData = [NSJSONSerialization dataWithJSONObject:body
61                                                         options:NSJSONWritingPrettyPrinted
62                                                         error:&error];
63
64 //Converte Data em String para impressao do JSON no Log
65 NSString *string = [[NSString alloc] initWithData:jsonData encoding:NSUTF8StringEncoding];
66 //Imprime JSON no Log.
67 NSLog(@"Testando os dados que irão subir %@", string);
68
69 //Adiciona o Data no Body(Corpo) da requisição
70 [request setHTTPBody:jsonData];
71
72 NSURLResponse *resposta;
73 //criando o NSData e fazendo um request, nesse nsdata teremos o retorno
74 //que o insert ocorreu com sucesso ou não
75 NSData *data = [NSURLConnection sendSynchronousRequest:request returningResponse:&resposta
76                                                         error:&error];
77
78 //transformando o NSDATA para NSDICTIONARY o NSJSONSerialization faz o Parser do JSON
79 NSDictionary *json = [NSJSONSerialization JSONObjectWithData:data options:kNilOptions
80                                                         error:&error];
81
82 NSLog(@"Exibindo os dados que retornaram da gravação"
83       "que pode ter sido com sucesso ou não: %@", json);
84
```


Implementando o Gravar

- Continuação.
- Execute novamente, veja que aparece a mensagem de gravação e a limpeza dos campos.

```
81
82     NSLog(@"Exibindo os dados que retornaram da gravação"
83           "que pode ter sido com sucesso ou não: %@", json);
84
85     //colhendo os dados (msg e status) que retornaram para duas variáveis
86     NSString *msg = [json objectForKey:@"mensagem"];
87     int status = [[json objectForKey:@"status"] intValue];
88
89
90     UIAlertView *alerta = [[UIAlertView alloc] initWithTitle:@"Aviso"
91                                                              message:msg
92                                                              delegate:self
93                                                              cancelButtonTitle:@"OK"
94                                                              otherButtonTitles:nil];
95
96     [alerta show];
97     //Limpando a tela caso a gravação tenha ocorrido com sucesso
98     if (status==1){
99         txtCaminho.text = @"";
100         txtRm.text = @"";
101         txtLocal.text = @"";
102         txtEstado.text = @"";|
103     }
```


Implementando o Gravar

- Command + R, preencha os dados e clique em Gravar.

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

Carrier 4:09 PM

Local

Estado

Caminho da Imagem

Rm do Aluno

Gravar

"12777"

Q W E R T Y U I O P

A S D F G H J K L

↑ Z X C V B N M ↵

123 😊 space return

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

Carrier 4:57 PM

Local

Estado

Caminho da Imagem

Rm do Aluno

Aviso

Inserido com sucesso.

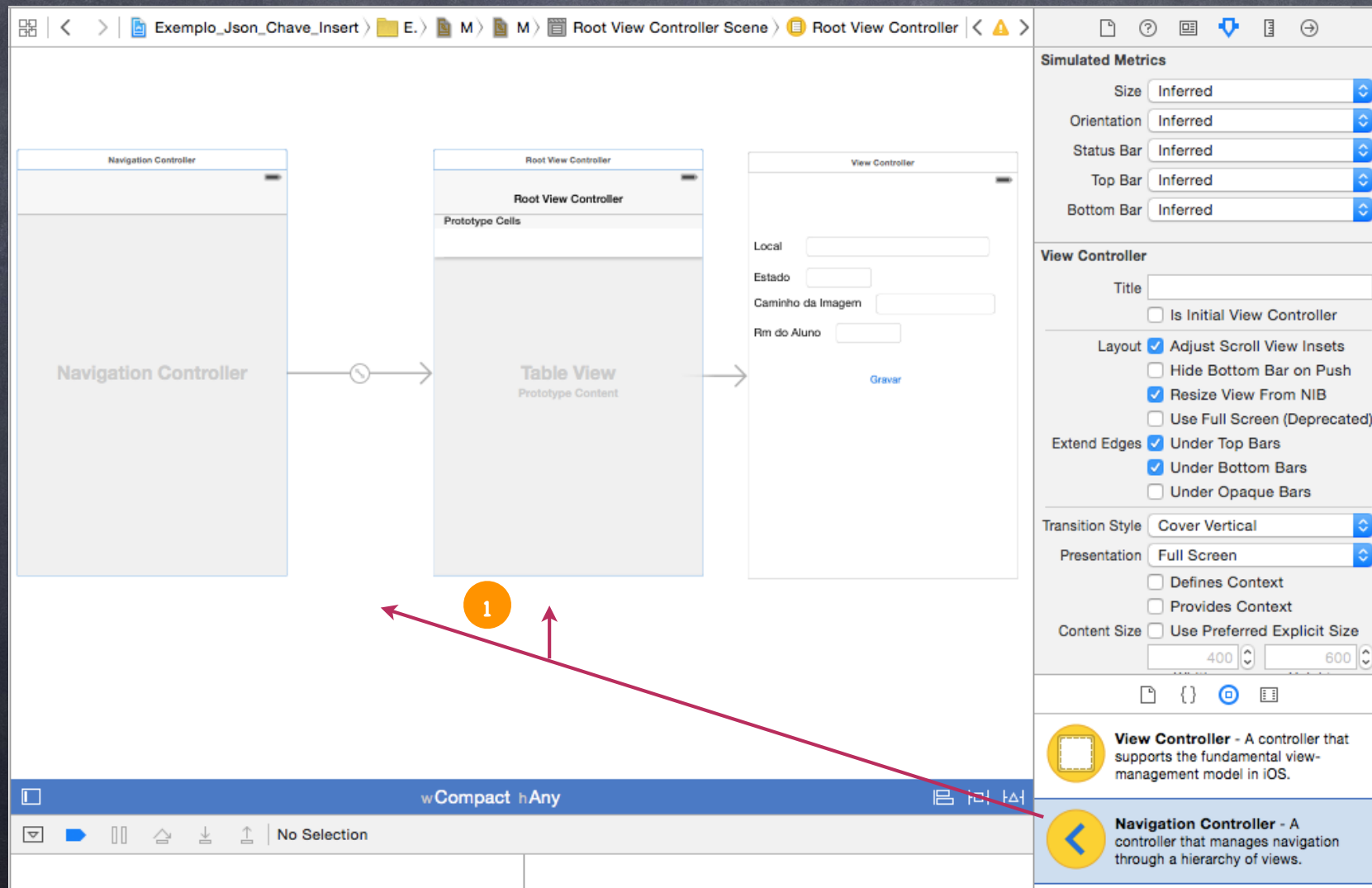
OK

Incluindo um TableViewController para listar os dados gravados

- Vamos incluir um Navigation Controller pois o mesmo já vem com um TableViewController.

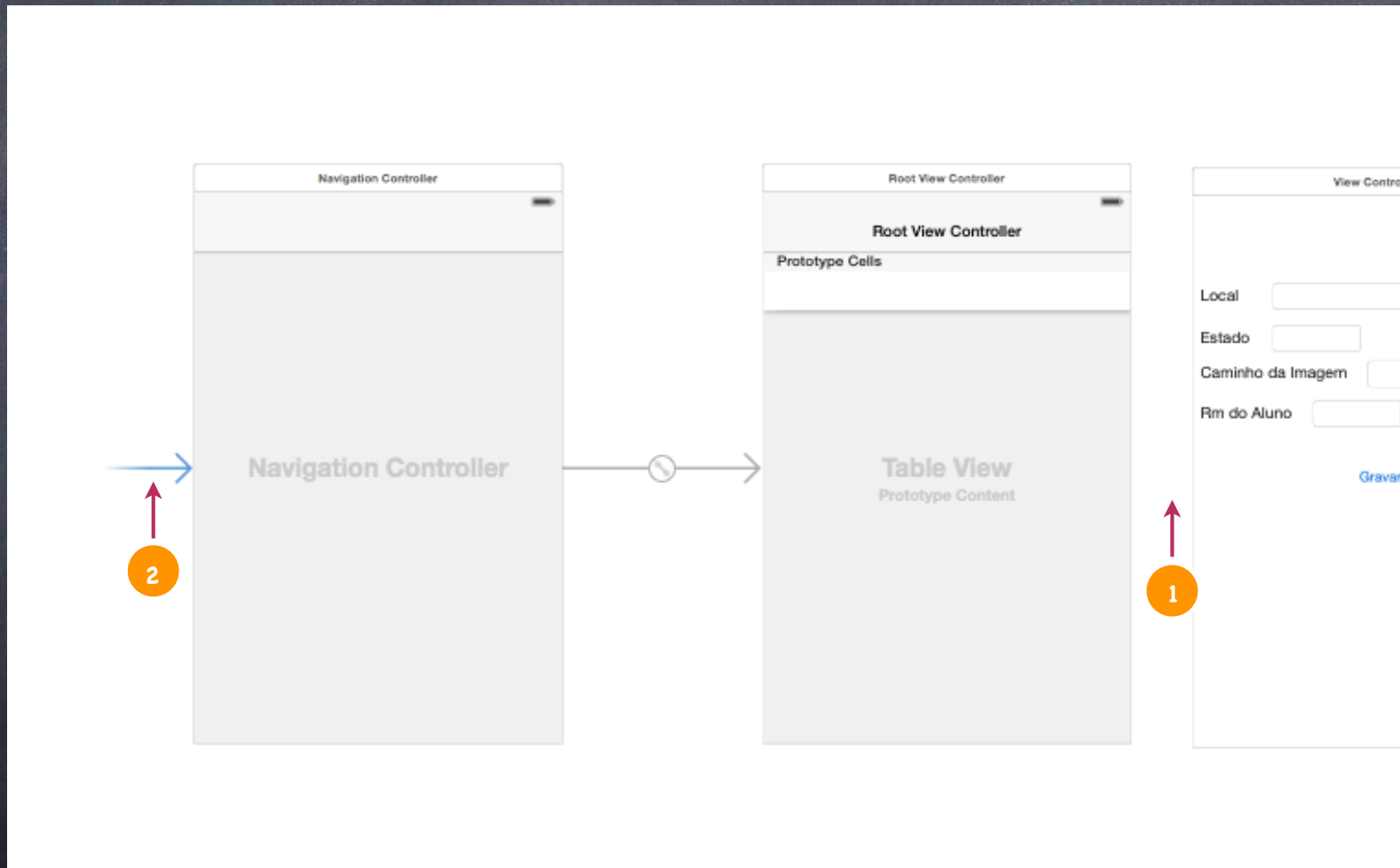
Preparando o Storyboard

- Clique 2x na área branca fora das telas no Storyboard para diminuir o zoom, arraste um UINavigationController para a área indicada(1).



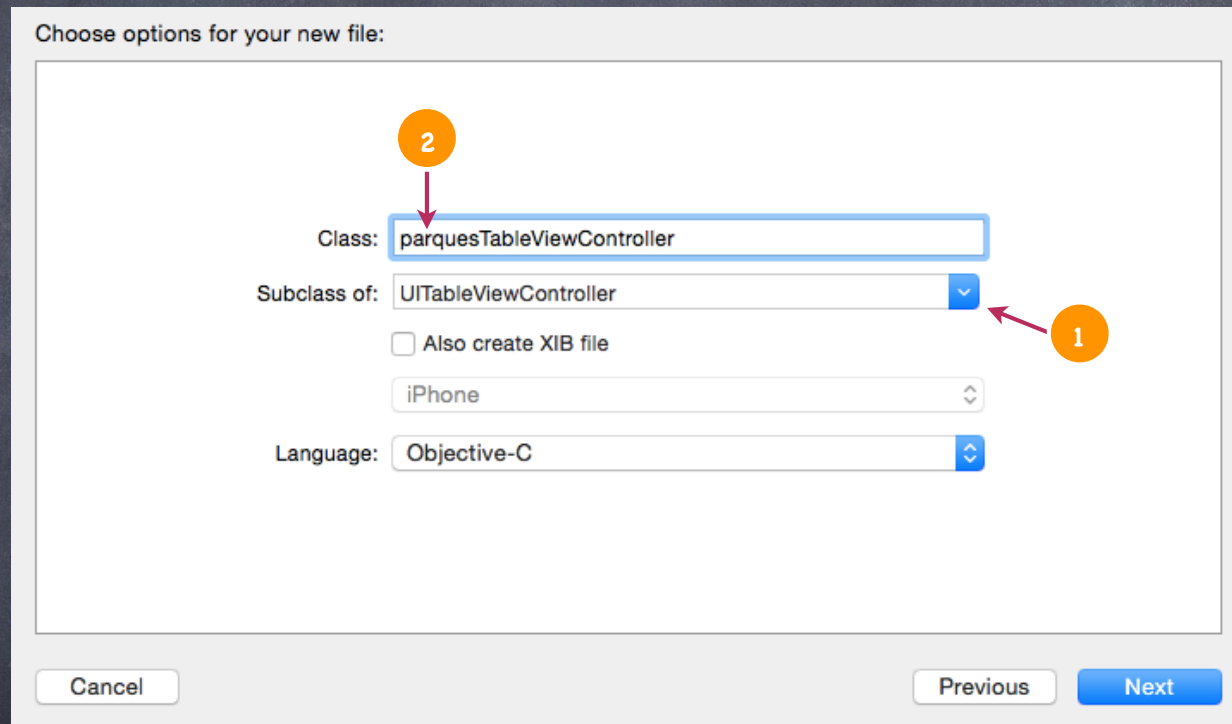
Preparando o Storyboard

- Arraste a seta do “Initial View Controller” (1) da ViewController que possui o botão gravar para a frente do Navigation Controller, dessa forma o navigation passa a ser a tela inicial.



Criando um nova classe

- Crie uma nova classe, filha de UITableViewController(1) chamada parquesTableViewController (2), (Command + N é o atalho).



Arquivo .h

- Insira as chaves e crie um array e um dicionário conforme as linha abaixo:

```
1  //
2  //  parquesTableViewController.h
3  //  Exemplo_Json_Chave_Insert
4  //
5  //  Created by agesandro scarpioni on 05/09/15.
6  //  Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.
7  //
8
9  #import <UIKit/UIKit.h>
10
11  @interface parquesTableViewController : UITableViewController{
12
13      NSMutableArray *jsonArray;
14      NSMutableDictionary *meuDicionario;
15  }
16
17  @end
18
```


Arquivo .m

- Preparando o request, definindo o método, setando a chave para o acesso e setando o RM para listar apenas os parques que o aluno incluiu, busque esse código na classe ViewController da página 9 e faça as adaptações necessárias.

```
17 - (void)viewDidLoad {
18     [super viewDidLoad];
19
20     // Uncomment the following line to preserve selection between presentations.
21     // self.clearsSelectionOnViewWillAppear = NO;
22
23     // Uncomment the following line to display an Edit button in the navigation bar for this view
24     // controller.
25     // self.navigationItem.rightBarButtonItem = self.editButtonItem;
26     NSString *stringCep = @"http://scarpioni.com/webservices/listar.php";
27
28     NSURL *url = [NSURL URLWithString:stringCep];
29
30     NSError *error;
31     //Criando uma requisição com a url informada
32     NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:url];
33     //Definindo o método como Post
34     [request setHTTPMethod:@"POST"];
35     //setando o tipo de conteúdo com json
36     [request setValue:@"application/json" forHTTPHeaderField:@"Content-Type"];
37     //setando uma chave para acesso aos dados
38     [request setValue:@"1234567890" forHTTPHeaderField:@"chave-api"];
39
40     NSMutableDictionary *body = [[NSMutableDictionary alloc] init];
41     //setar seu rm para listar apenas os parques que vc cadastrou
42     [body setValue:@"12777" forKey:@"rm"];
```


Arquivo .m

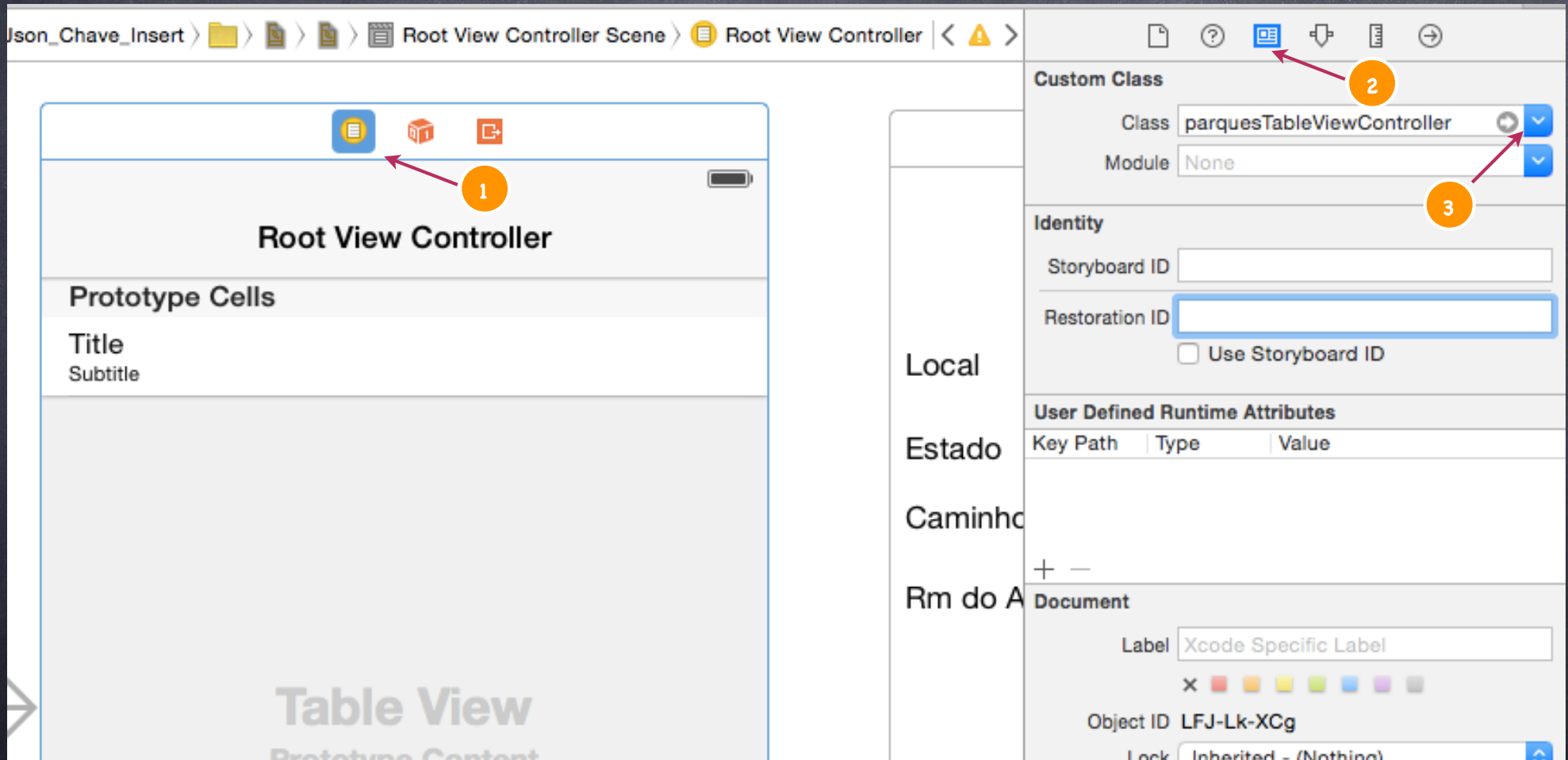
- Serializa o dicionário para Data, adiciona o data no body do request, executa o request e transforma o data para dicionário.

```
42
43 //Serializa o Dictionary para Data
44 NSData *jsonData = [NSJSONSerialization dataWithJSONObject:body
45                                     options:NSJSONWritingPrettyPrinted
46                                     error:&error];
47
48 //Converte Data em String para impressao do JSON no Log
49 NSString *string = [[NSString alloc] initWithData:jsonData
50                                     encoding:NSUTF8StringEncoding];
51
52 //Imprime JSON no Log.
53 NSLog(@"%@", string);
54
55 //Adiciona o Data no Body(Corpo) da requisição
56 [request setHTTPBody:jsonData];
57
58 NSURLResponse *resposta;
59 //criando o NSData e fazendo um request
60 NSData *data = [NSURLConnection sendSynchronousRequest:request
61                                     returningResponse:&resposta
62                                     error:&error];
63
64 //transformando o NSData para NSDictionary o NSJSONSerialization faz o Parser do JSON
65 jsonArray = [NSJSONSerialization JSONObjectWithData:data
66                                     options:kNilOptions
67                                     error:&error];
68
69 //testes
70 NSLog(@"%@", jsonArray); //exibindo o array para teste
71 meuDicionario = [jsonArray objectAtIndex:0]; //passando para o dicionario o item 0 do array
72 NSLog(@"Nome do parque para teste %@", [meuDicionario objectForKey:@"nome"]);
```


Identity Inspector

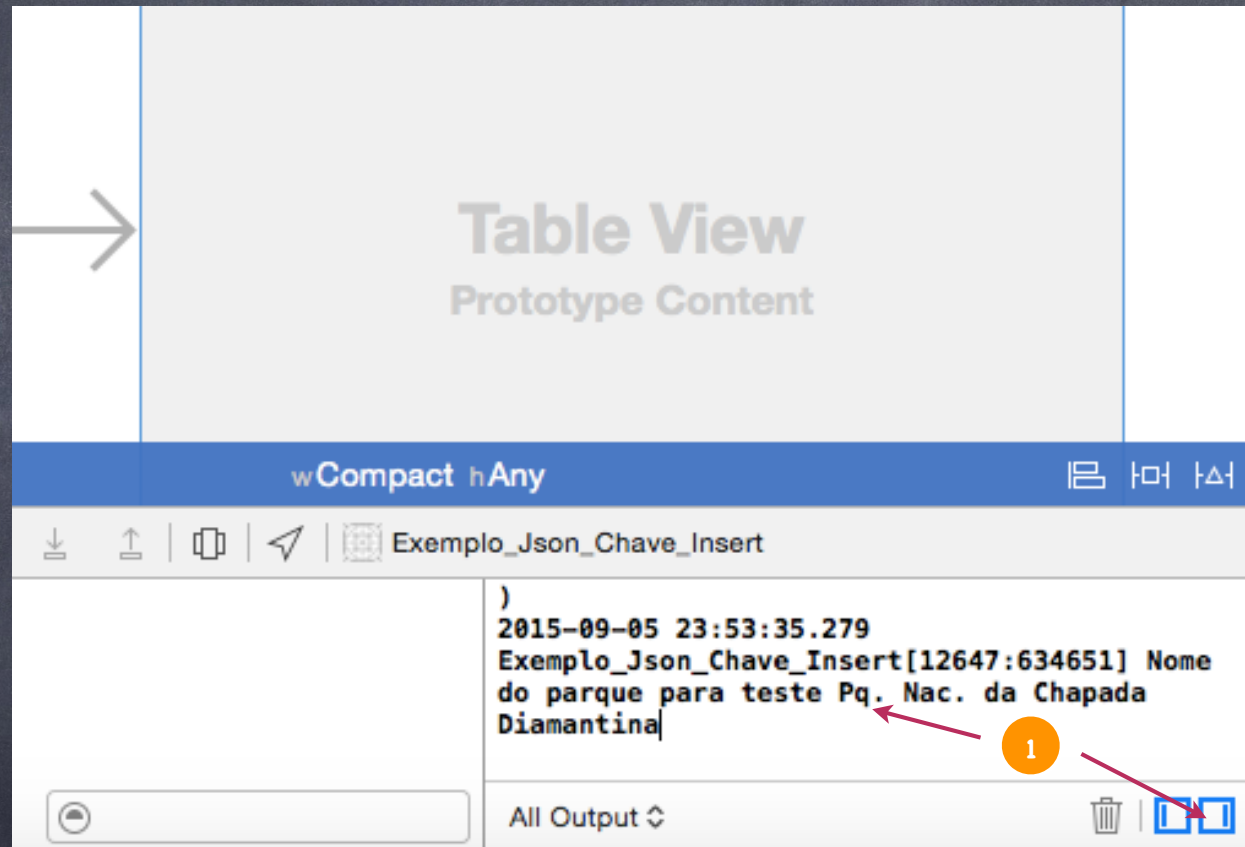
FIAP

- Vamos relacionar essa controle a view criada, no StoryBoard selecione o TableViewController (1) , vá até o Identity Inspector(2) e selecione a classe "Dona" da View em Class(3).



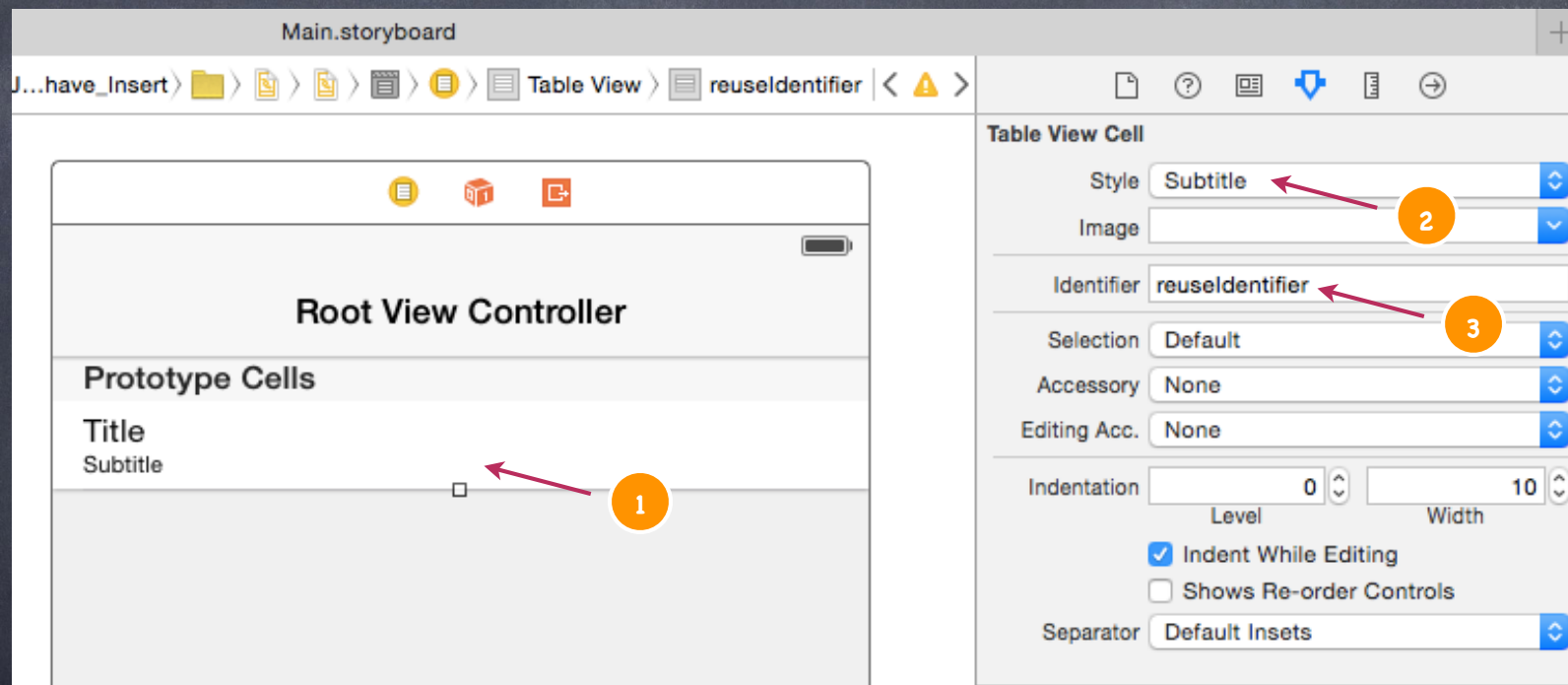
Testando

- Execute o código e observe no console se as informações de NSLog Json irão aparecer.



Configurando a Cell


- Selecione a TableViewCell (1), escolha o estilo para Subtitle (2) e informe no Identifier (3) o nome da célula.



Implementando o Data Source

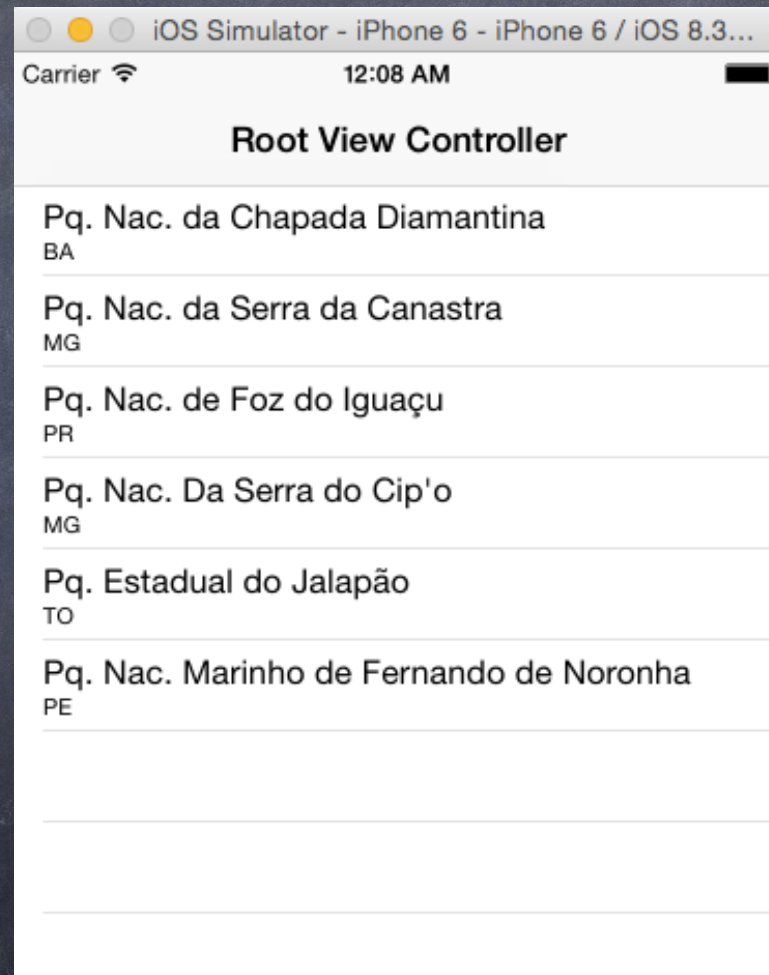
- Lembre-se de clicar 2x sobre o nome "reuseIdentifier" para aceitar a sugestão de nome, a string tem que ficar em vermelho (1), depois configure a célula.

```
78
79 #pragma mark - Table view data source
80
81 - (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {
82     // Return the number of sections.
83     return 1;
84 }
85
86 - (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
87 {
88     // Return the number of rows in the section.
89     return [jsonArray count];
90 }
91
92 - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:
93     (NSIndexPath *)indexPath {
94     UITableViewCell *cell = [tableView
95         dequeueReusableCellWithIdentifier:@"reuseIdentifier" forIndexPath:indexPath];
96
97     // Configure the cell...
98     meuDicionario = [jsonArray objectAtIndex:indexPath.row];
99
100     cell.textLabel.text = [meuDicionario objectForKey:@"nome"];
101     cell.detailTextLabel.text = [meuDicionario objectForKey:@"estado"];
102     return cell;
103 }
```

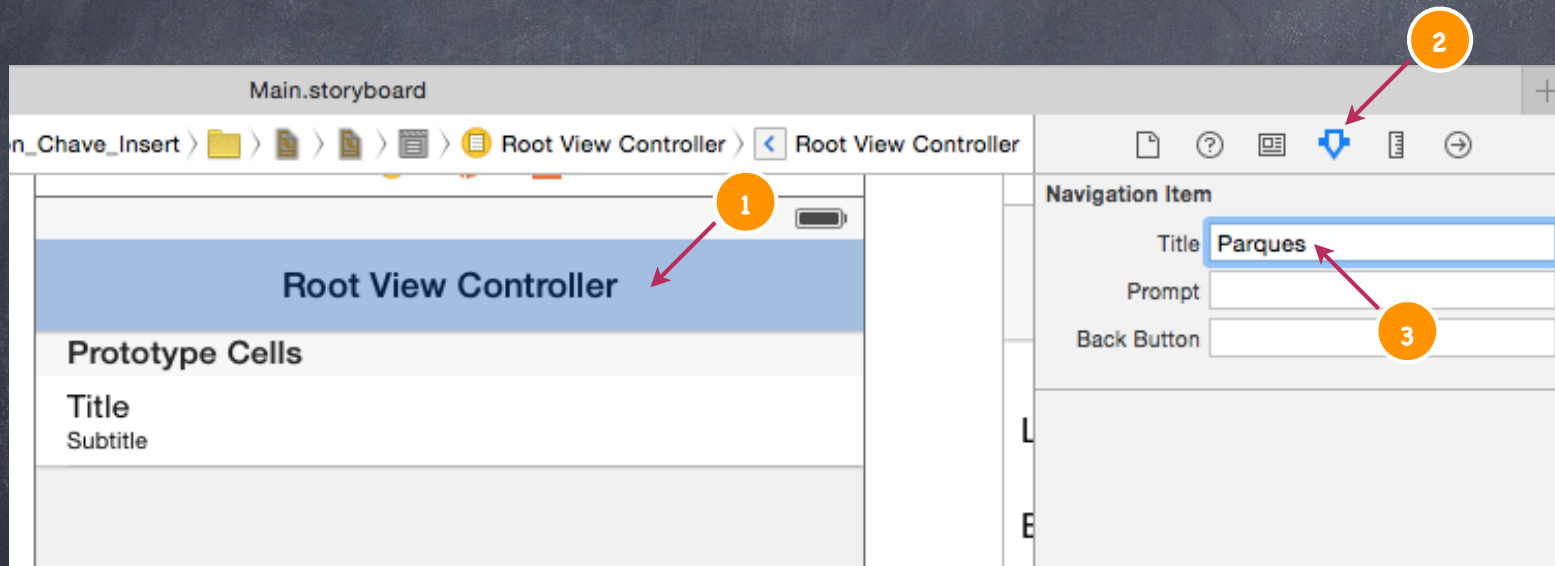


Testando

- Execute o código e observe os dados no TableView

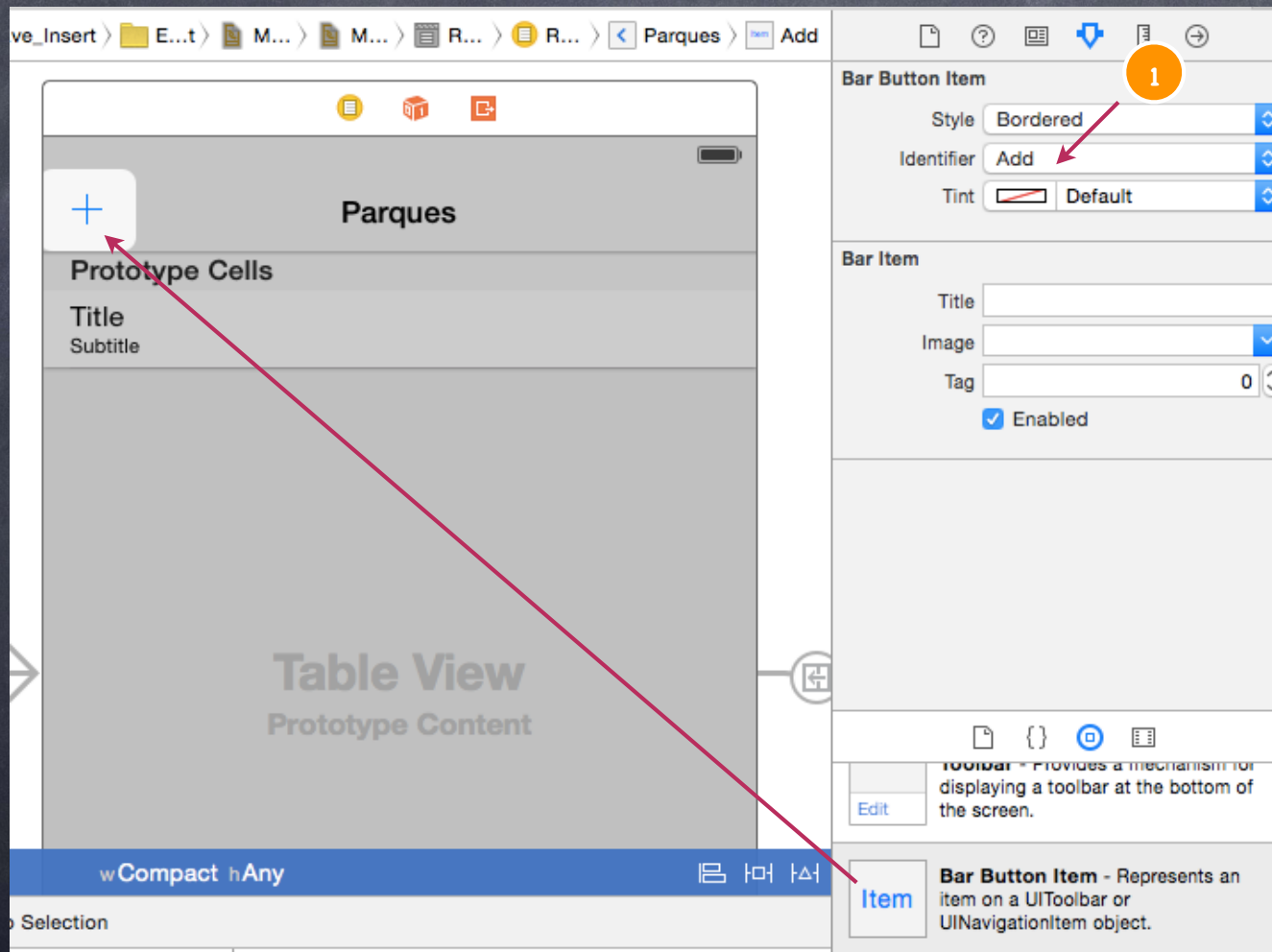


Trocar o Nome do Navigation Item



Incluindo um Bar Button Item no Navigation Item

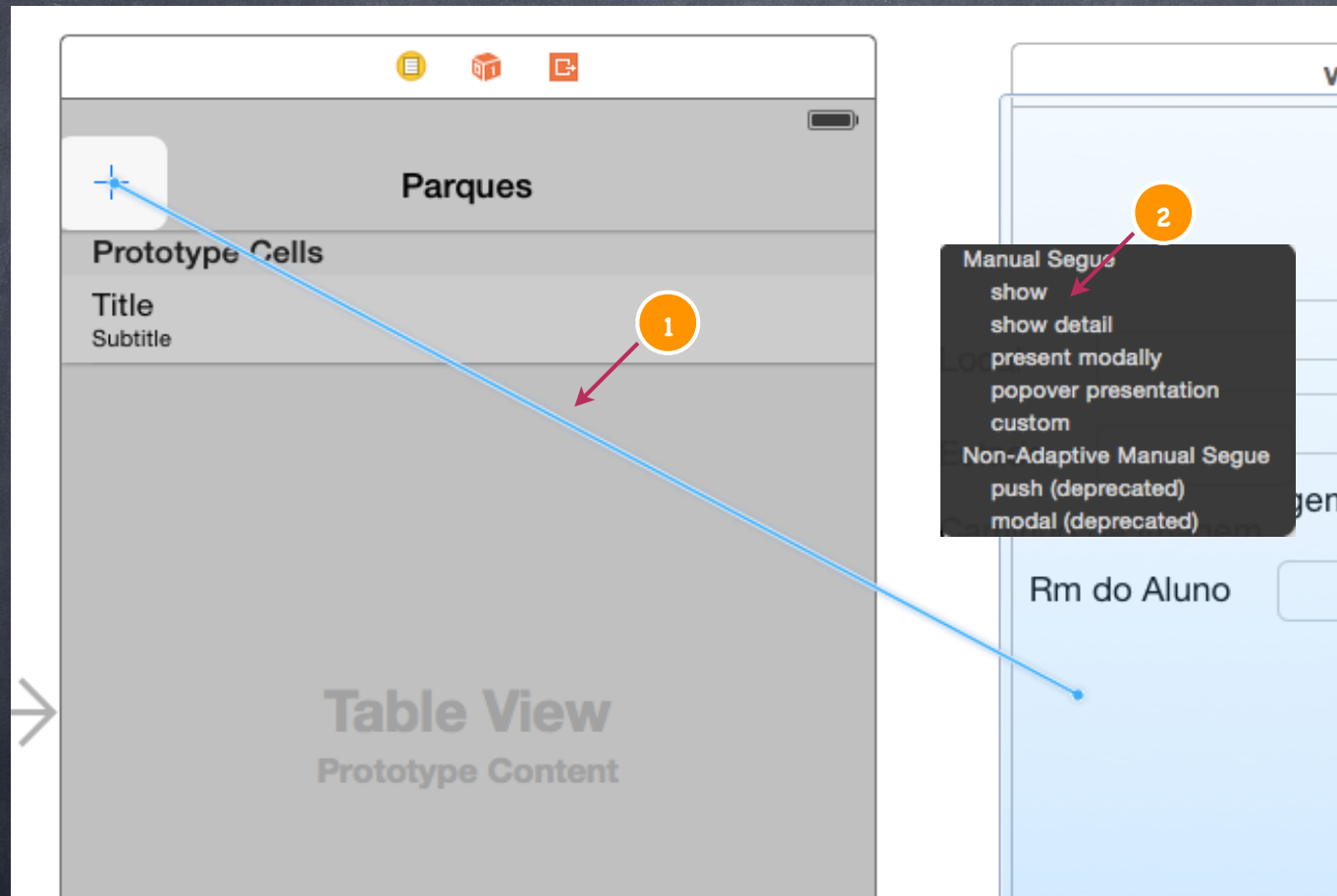
- Em Attributes Inspector altere o Identifier para Add(1).



Obs: Ao arrastar o Bar Button, espere até que uma área azul apareça ao lado da palavra Parques, depois solte o botão

Bar Button Item

- Ligando o botão Bar Button Item ao View Controller (1) e escolha Show (2).



Bar Button Item


- Em parques TableViewController, crie após o viewDidLoad um novo método chamado viewWillAppear (1).

```
63 //transformando o NSData para NSDictionary o NSJSONSeriali
64 jsonArray = [NSJSONSerialization JSONObjectWithData:data
65                                     options:kNilO
66                                     error:&erro
67 //testes
68 NSLog(@"%@", jsonArray); //exibindo o array para teste
69 meuDicionario = [jsonArray objectAtIndex:0]; //passando pa
70 NSLog(@"Nome do parque para teste %@", [meuDicionario obj
71 }
72 }
73
74 -(void) viewWillAppear:(BOOL)animated{
75     |
76 }
77
78 - (void) didReceiveMemoryWarning {
79     [super didReceiveMemoryWarning];
80     // Dispose of any resources that can be recreated.
81 }
82
83 #pragma mark - Table view data source
```


Bar Button Item

- Recorte toda a programação do viewDidLoad e cole no viewWillAppear, incluindo a linha: [self.tableView reloadData] (1).


```
71 //testes
72 NSLog(@"%@", jsonArray); //exibindo o array para teste
73 meuDicionario = [jsonArray objectAtIndex:0]; //passar
74 NSLog(@"Nome do parque para teste %@", [meuDicionario
75 | [self.tableView reloadData];
76 }
77
78 - (void)didReceiveMemoryWarning {
79     [super didReceiveMemoryWarning];
80     // Dispose of any resources that can be recreated.
81 }
82
```



Método Gravar

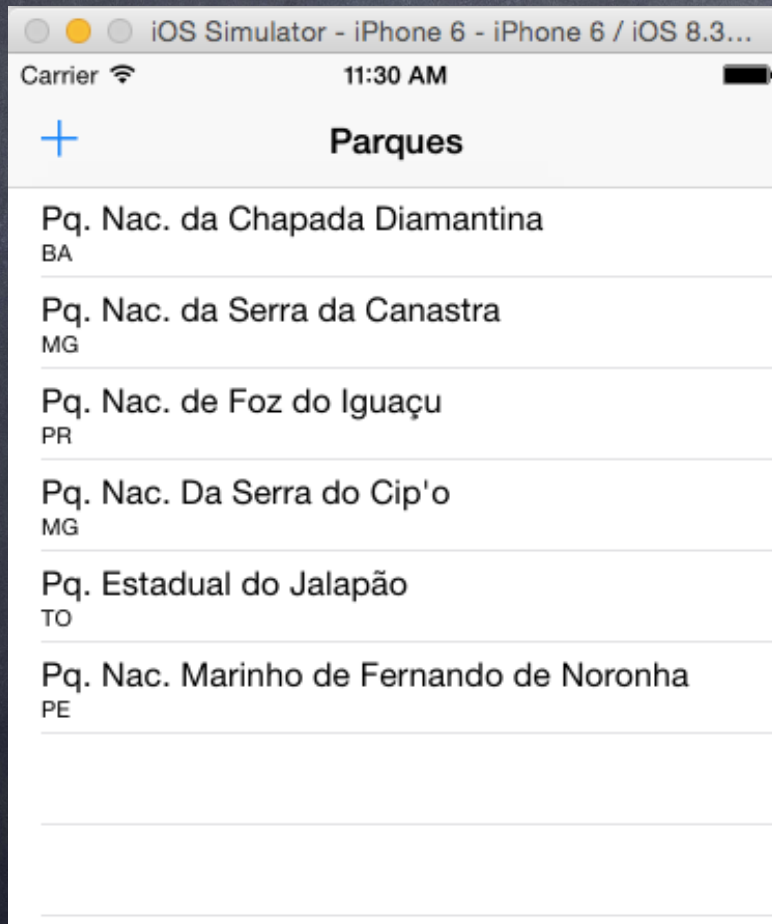
- No viewController, ao final da linha do método gravar, inclua a linha: `[self.navigationController popViewControllerAnimated:YES]` (1), essa chamada serve para fechar a pop que foi aberta.

```
89
90     if (status==1){
91         txtCaminho.text = @"";
92         txtRm.text = @"";
93         txtLocal.text = @"";
94         txtEstado.text = @"";
95         [self.navigationController popViewControllerAnimated:YES];
96     }
97
```



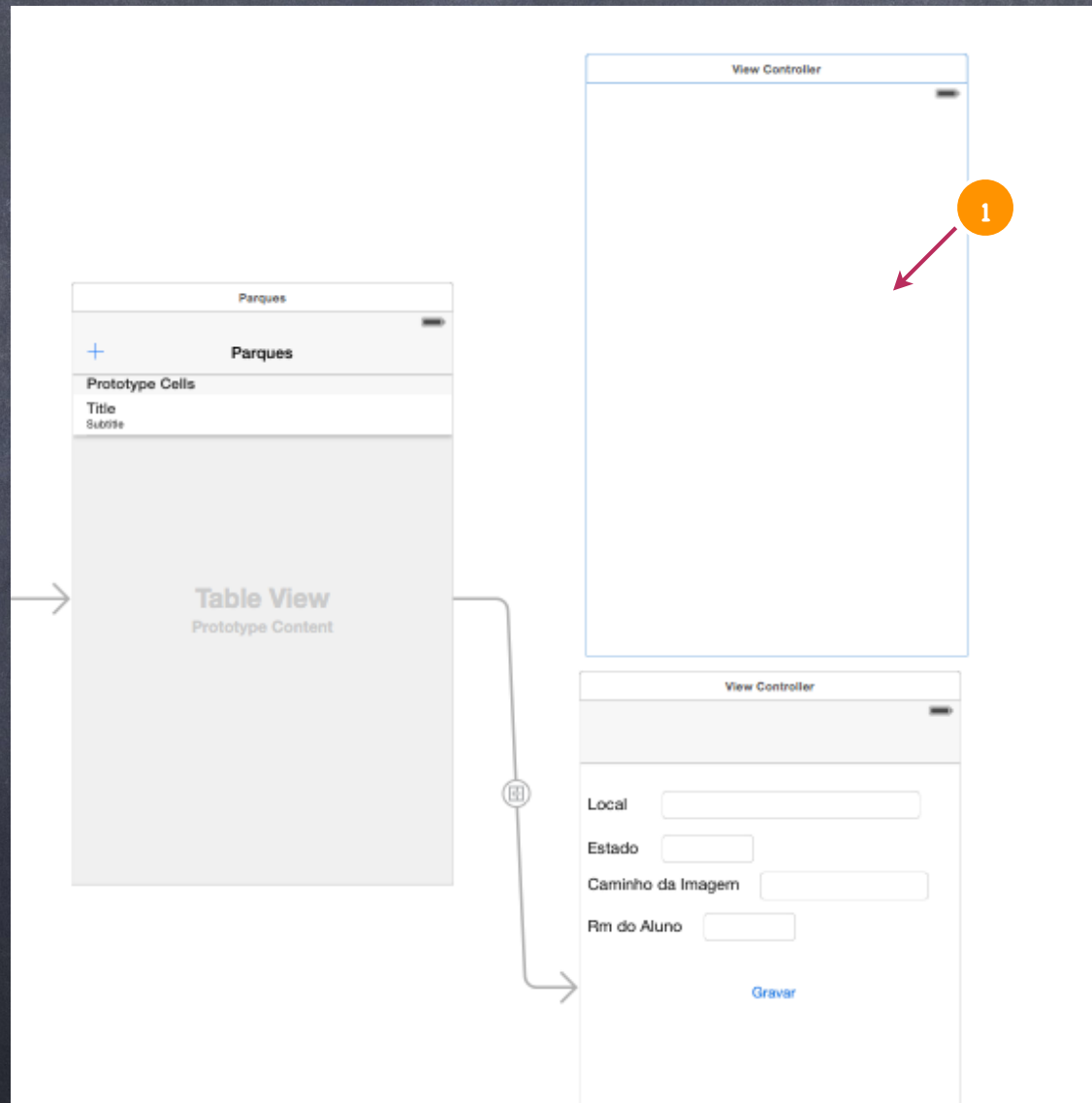
Testando

- Execute o código e observe o Bar Button Item, cadastre os dados e note que a tela fecha e o resultado já aparece na TableView.



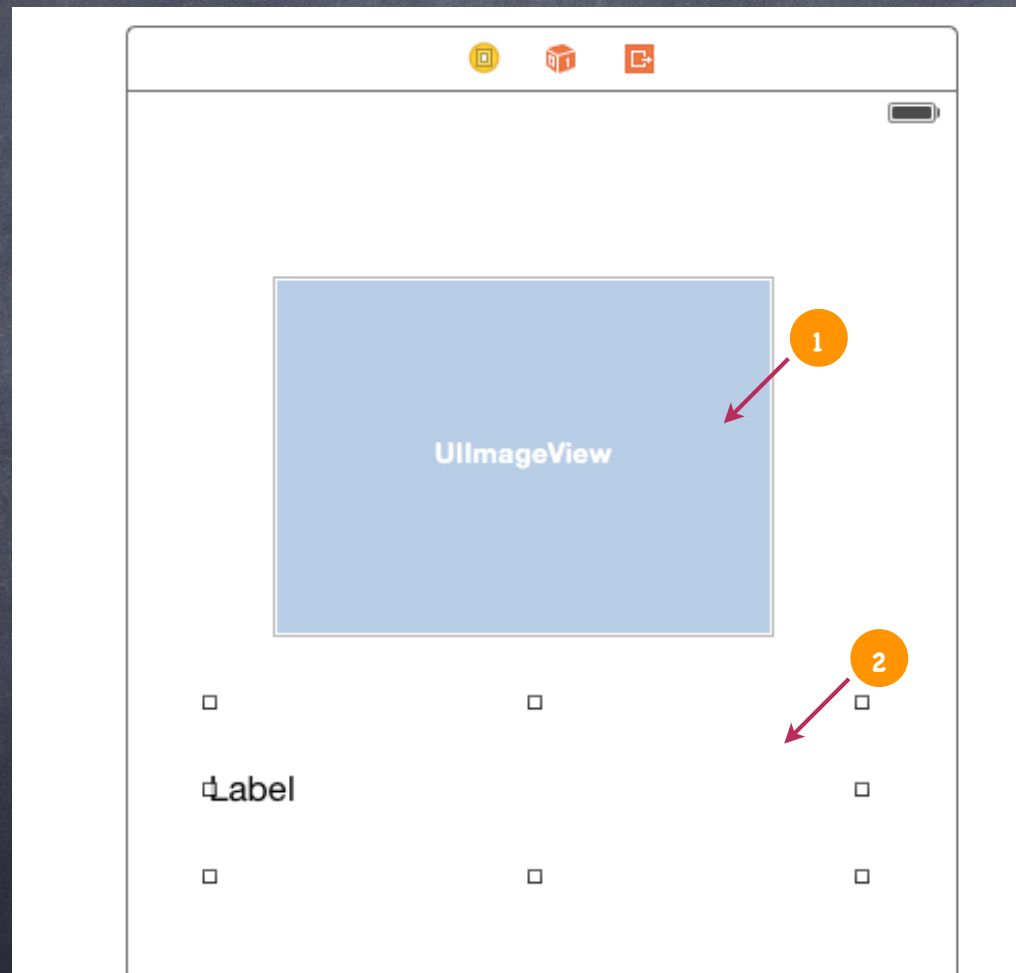
Chamar Segue Via Código

- Diminua o zoom como mostra a figura abaixo e inclua uma nova ViewController (1).



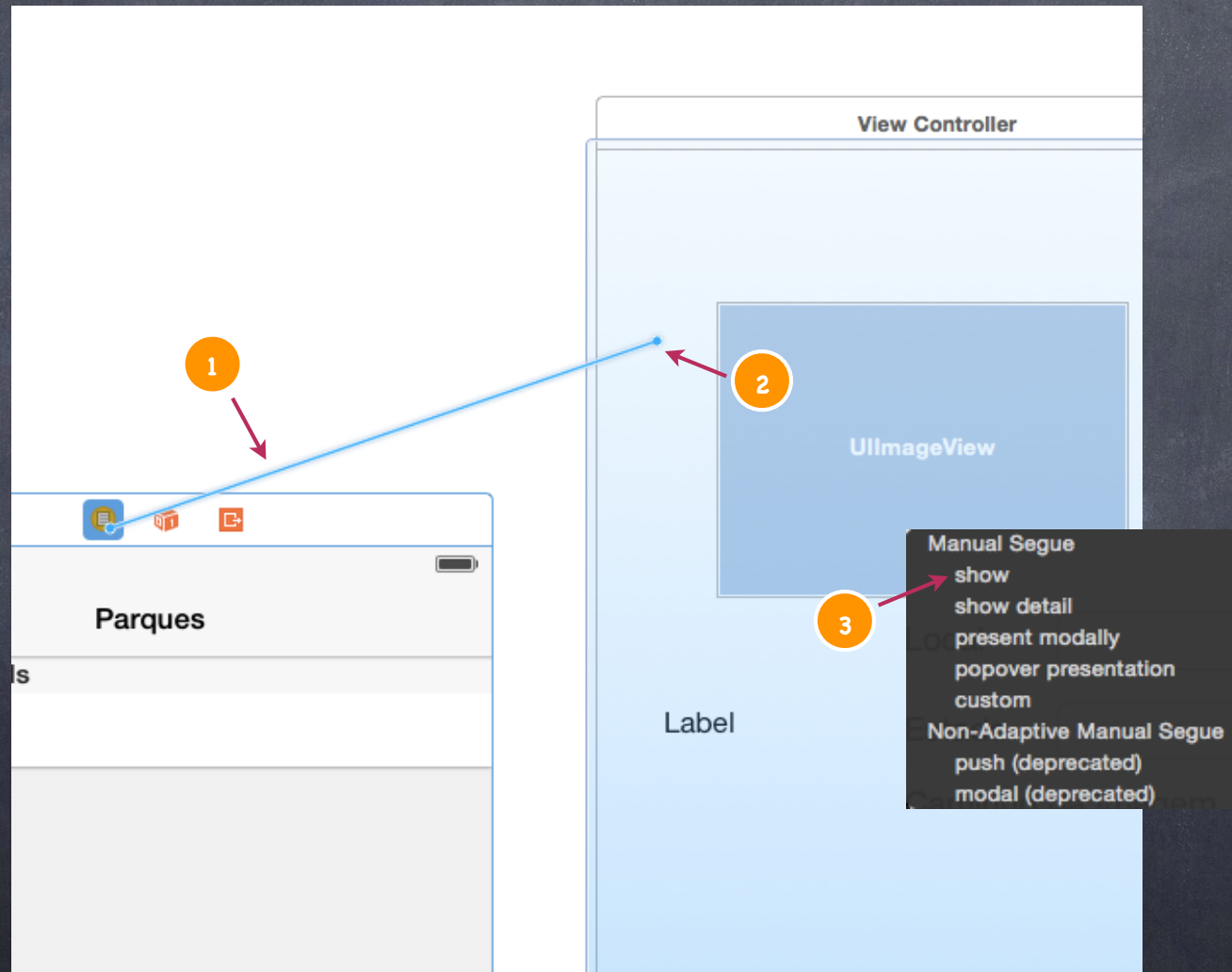
Chamar Segue Via Código

- Retorne o zoom para a posição inicial e nessa nova tela inclua um UIImageView (1) e um Label (2).



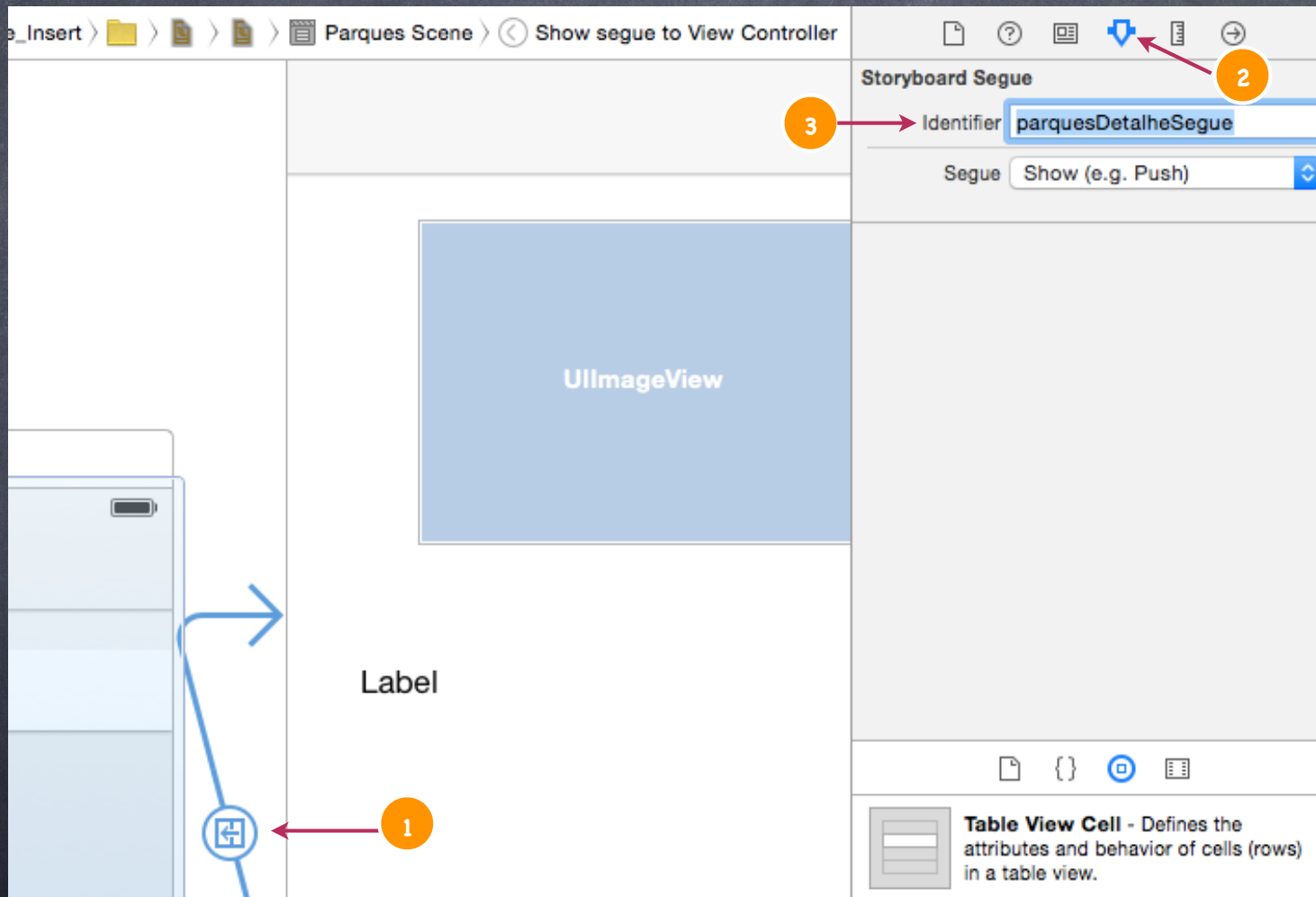
Chamar Segue Via Código

- Passo 1: Ligar Controller com controller conforme a imagem abaixo e escolher show.



Chamar Segue Via Código

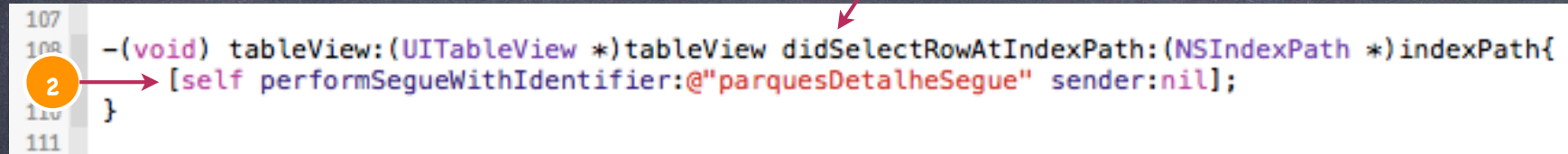
- Passo 2: Selecionar a Segue (1) e em Attributes Inspector (2), informar o nome da segue no campo Identifier(3).



Chamar Segue Via Código

FIAP

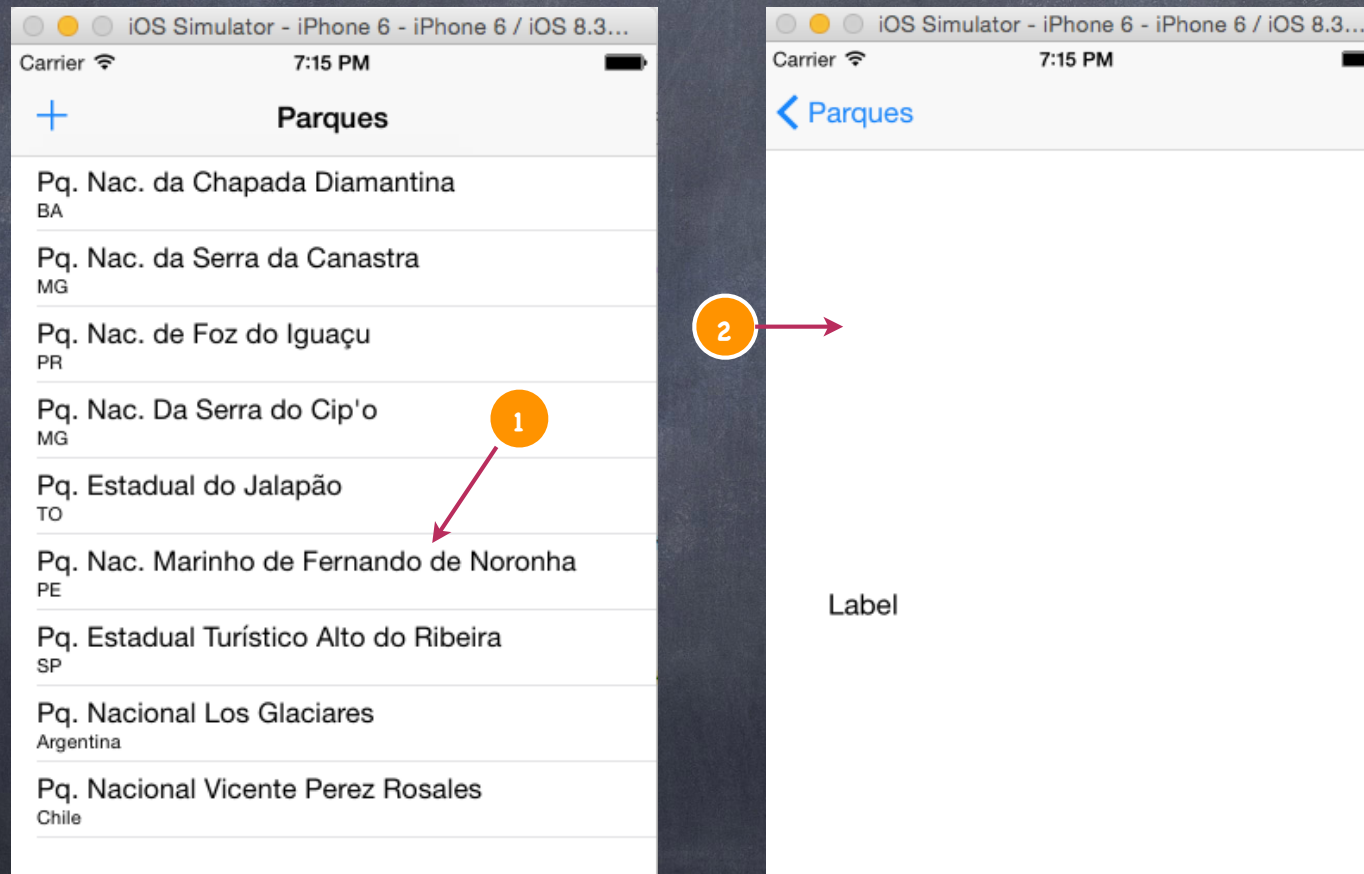
- Passo 3: Em tableViewController inclua o método didSelectRowAtIndexPath(1), implemente a linha indicada(2).



```
107  
108  
109 -(void) tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath{  
110     [self performSegueWithIdentifier:@"parquesDetalheSegue" sender:nil];  
111 }
```

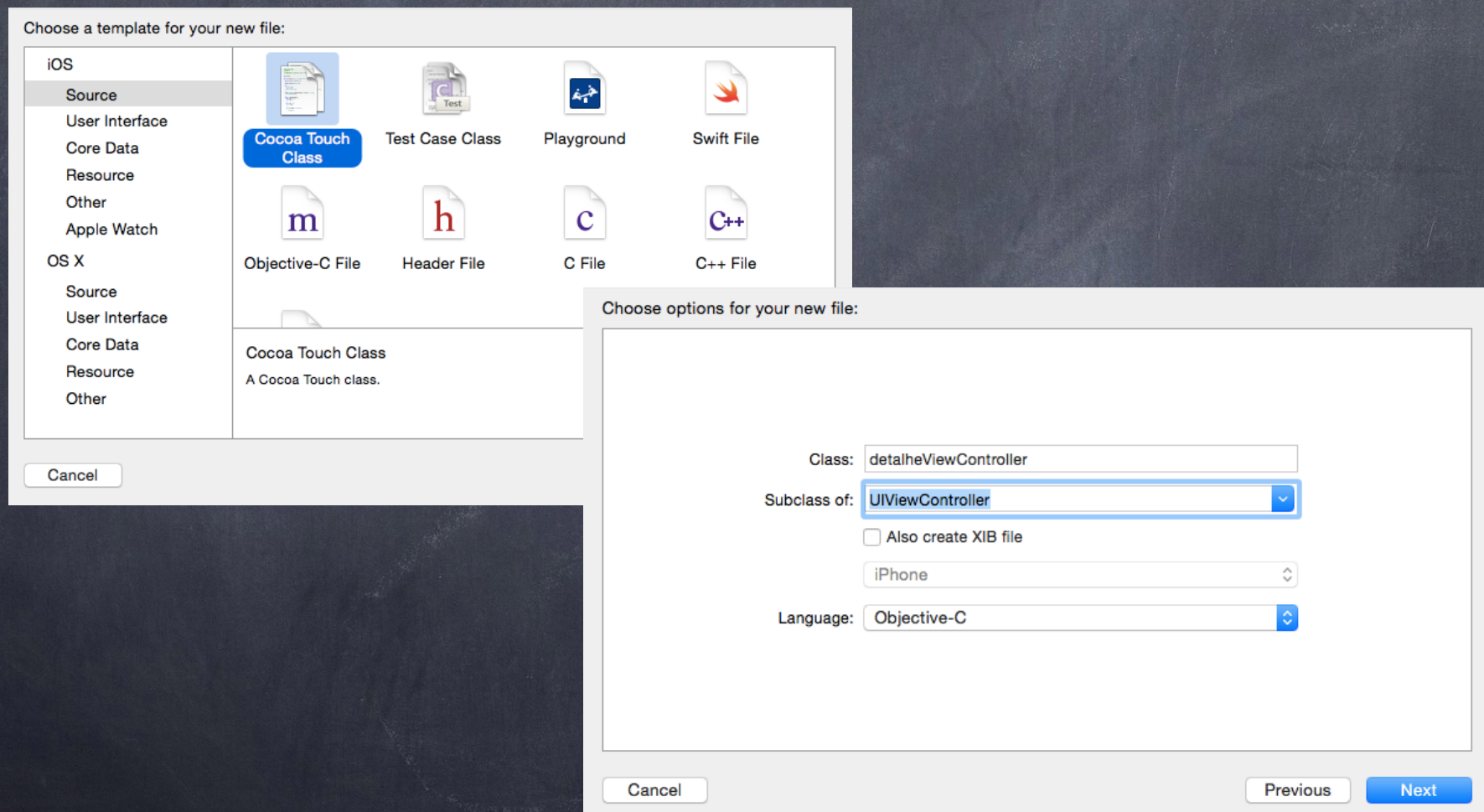

Testando

- Execute seu projeto e observe que ao clicarmos na TableView uma tela será aberta via código.



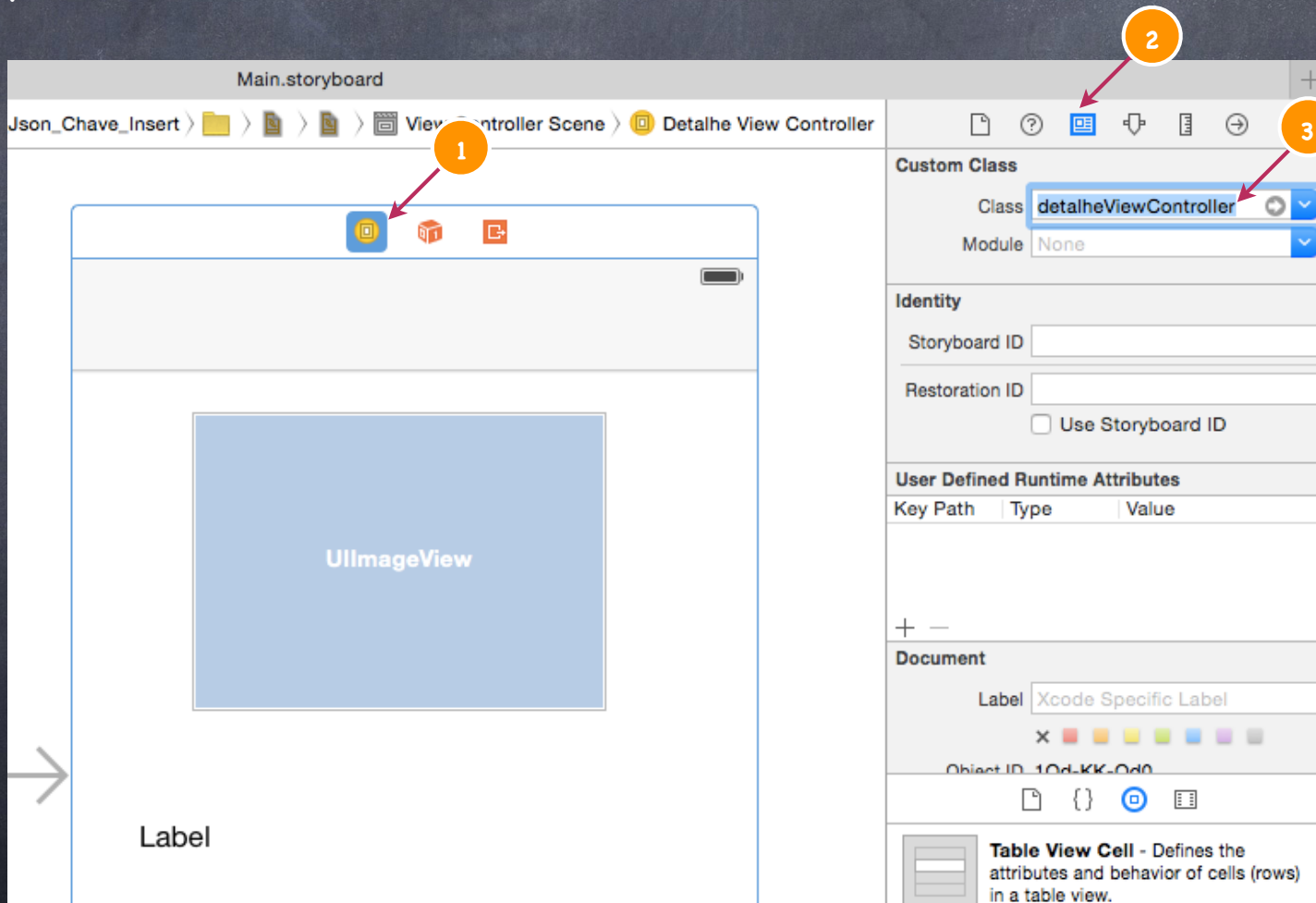
Passando os dados de uma View para outra

- Crie mais uma classe (command + N) chamada detalheViewController, subclasse de ViewController.



FIAP

- Informe qual classe é "dona" da View, selecione a View(1) no botão Identity Inspector(2) e selecione detalheViewControler(3).



Passando os dados de uma View para outra


- Crie os outlets para a imagem e o label, além de criar 2 atributos para os mesmos campos conforme o exemplo abaixo:

```
1 //  
2 // detalheViewController.h  
3 // Exemplo_Json_Chave_Insert  
4 //  
5 // Created by agesandro scarpioni on 07/09/15.  
6 // Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.  
7 //  
8  
9 #import <UIKit/UIKit.h>  
10  
11 @interface detalheViewController : UIViewController  
12  
13  
14 @property (weak, nonatomic) IBOutlet UIImageView *ImgImagem;  
15 @property (weak, nonatomic) IBOutlet UILabel *lblParque;  
16  
17 @property (nonatomic,retain) NSString *nomeParque;  
18 @property (nonatomic,retain) UIImage *imagem;  
19 |  
20  
21 @end  
22
```


Passando os dados de uma View para outra

- Faça no viewDidLoad a programação para passar os atributos para os respectivos outlets.

```
1  //
2  //  detalheViewController.m
3  //  Exemplo_Json_Chave_Insert
4  //
5  //  Created by agesandro scarpioni on 07/09/15.
6  //  Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.
7  //
8
9  #import "detalheViewController.h"
10
11 @interface detalheViewController ()
12
13 @end
14
15 @implementation detalheViewController
16
17 - (void)viewDidLoad {
18     [super viewDidLoad];
19     _lblParque.text = _nomeParque;
20     _ImgImagem.image = _imagem;
21 }
22
23
```



Passando os dados de uma View para outra

- Na classe parquesTableViewController faça o import de detalheViewController(1).

```
1  //
2  //  parquesTableViewController.m
3  //  Exemplo_Json_Chave_Insert
4  //
5  //  Created by agesandro scarpioni on 05/09/15.
6  //  Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.
7  //
8
9  #import "parquesTableViewController.h"
10 #import "detalheViewController.h" ← 1
11
12
13 @interface parquesTableViewController ()
14
15 @end
```


Passando os dados de uma View para outra

- Crie uma variável na classe parquesTableViewController para representar o índice da cell clicada(1).

```
1 //
2 // parquesTableViewController.h
3 // Exemplo_Json_Chave_Insert
4 //
5 // Created by agesandro scarpioni on 05/09/15.
6 // Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.
7 //
8
9 #import <UIKit/UIKit.h>
10
11 @interface parquesTableViewController : UITableViewController{
12
13     NSMutableArray *jsonArray;
14     NSMutableDictionary *meuDicionario;
15     NSInteger indice; ← 1
16     |
17 }
18
19 @end
20
```

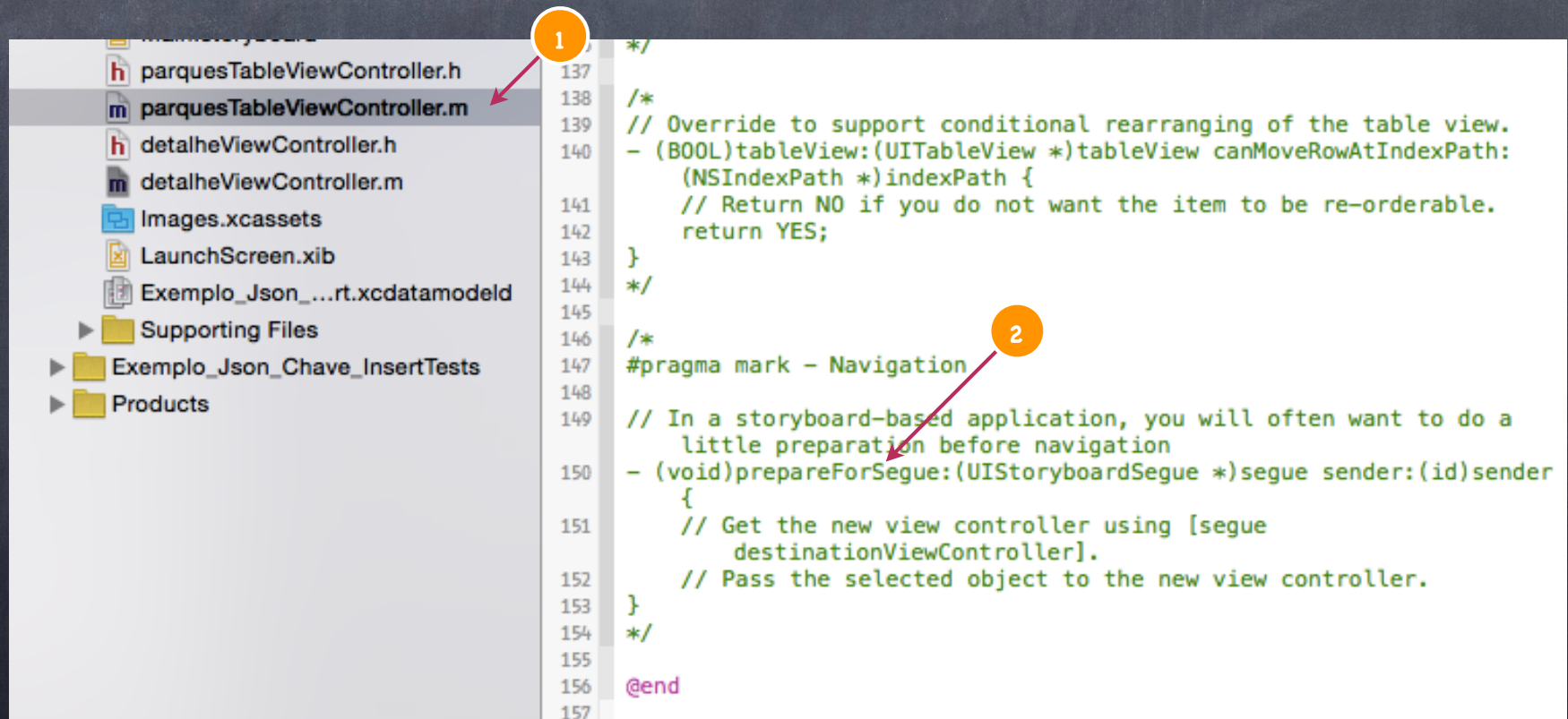

Passando os dados de uma View para outra

- Atualize a variável índice no didSelectRowAtIndexPath da classe parquesTableViewController(1).

```
109  
110 -(void) tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath{  
111     indice = indexPath.row; ← 1  
112     [self performSegueWithIdentifier:@"parquesDetalheSegue" sender:nil];  
113 }  
114
```


Passando os dados de uma View para outra

- Procure na classe parquesTableViewController(1) o método prepareForSegue(2) esse método fica no final da classe, retire o comentário e faça a implementação conforme o próximo slide.



Passando os dados de uma View para outra

- Implemente as linhas abaixo para exibir o nome do parque quando clicarmos na cell.

```
151
152 #pragma mark - Navigation
153
154 // In a storyboard-based application, you will often want to do a little preparation before navigation
155 - (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
156     if([segue.identifier isEqualToString:@"parquesDetalheSegue"]){
157         detalheViewController *vc = segue.destinationViewController;
158         meuDicionario = [jsonArray objectAtIndex:indice];
159         vc.nomeParque = [meuDicionario objectForKey:@"nome"];
160         //agora faça as alterações necessárias para aparecer as fotos
161     }
162 }
163
```


Atividade

- Faça a programação para exibir a imagem do parque no UIImageView.

