



POLITÉCNICA

ETSIT  
UPM

*dit*  
UPM

# Desarrollo de Apps para iOS

## Demo: Pokedesk

IWEB,LSWC 2013-2014  
Santiago Pavón

ver: 2013.10.21

# Descripción de la Demo

- Desarrollar una aplicación con dos pantallas:
  - Se navega entre ellas usando un Navigation Controller.
- La primera pantalla muestra una tabla con las razas de los pokemons
  - Usando secciones para agruparlas por tipo.
- Al seleccionar una raza de la tabla cambiamos a la segunda pantalla.
  - Muestra un navegador web que se descarga a una página con información sobre la raza seleccionada.
- Los datos se cargarán desde un fichero de propiedades (plist).



# Crear el Proyecto

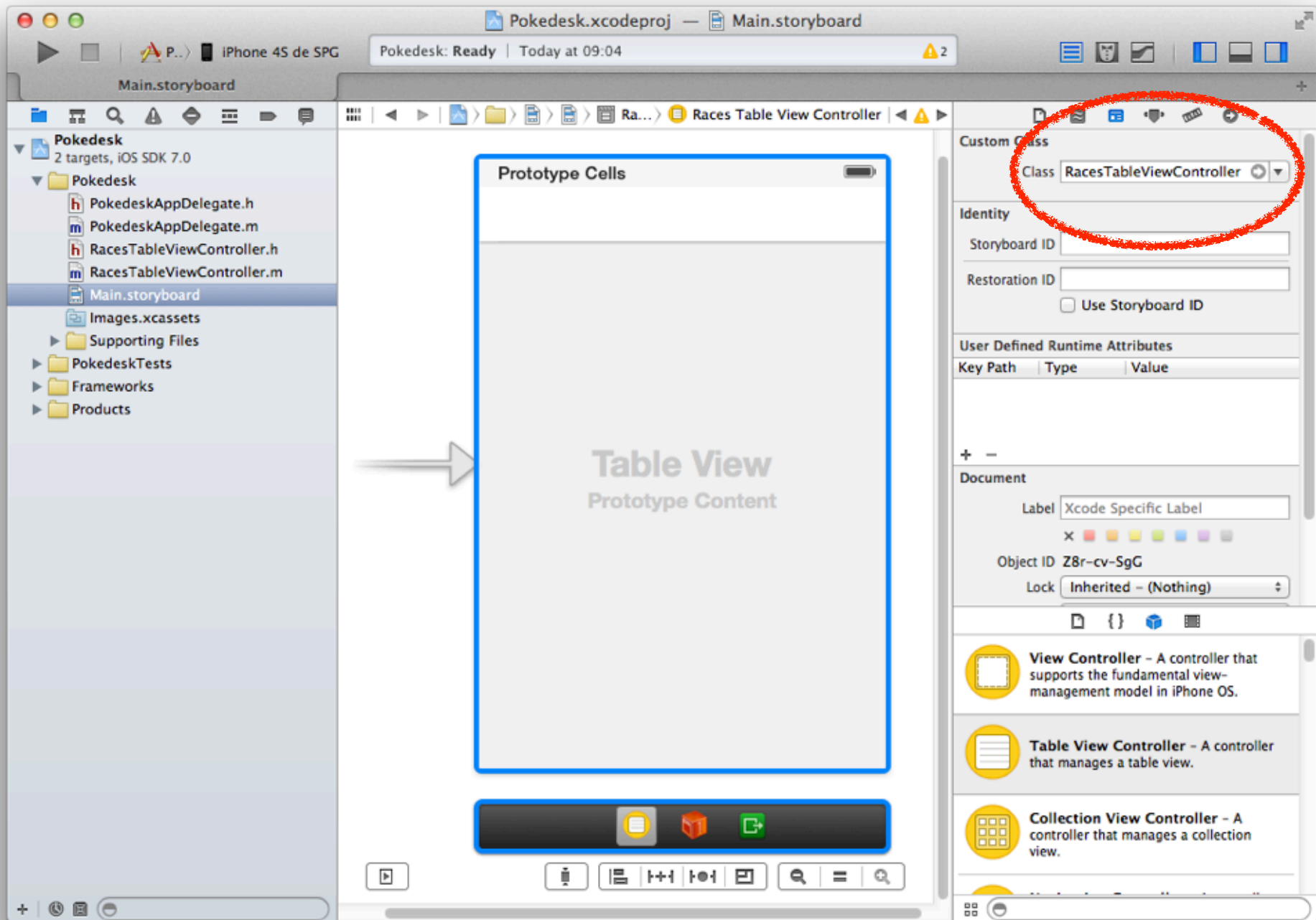
- Lanzar Xcode y crear un proyecto nuevo.
  - Plantilla: Single View Application
  - Nombre: Pokedesk
  - Device: iPhone

# Crear la Tabla de Razas

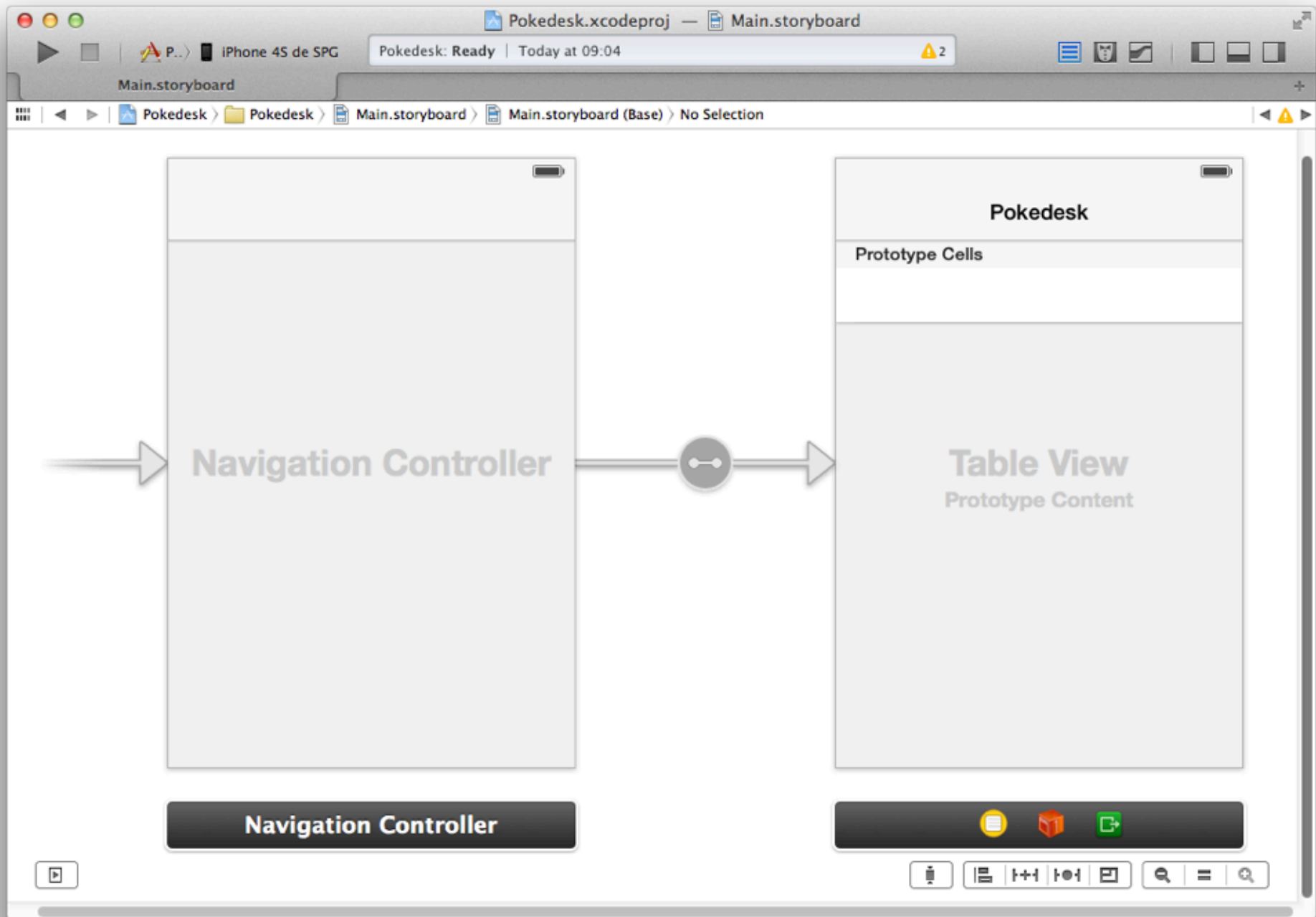
- **Borrar** los siguientes elementos que creó la plantilla:
  - Usando Interface Builder, eliminar el ViewController del storyboard.
  - Los dos ficheros `ViewController.h` y `.m`.
- Añadir al storyboard un objeto **UITableViewController**, arrastrándolo desde la librería de objetos.
  - En él mostraremos las razas de los pokemons.
- Crear una clase nueva, llamada **RacesTableViewController**, para la UITableViewController que acabamos de añadir al Storyboard.

*New File > New > File > iOS Cocoa Touch > Objective-C class*

  - Esta clase debe derivar de **UITableViewController**.
- Con el inspector de identidad del Interface Builder cambiar la clase del **UITableViewController** a **RacesTableViewController**.

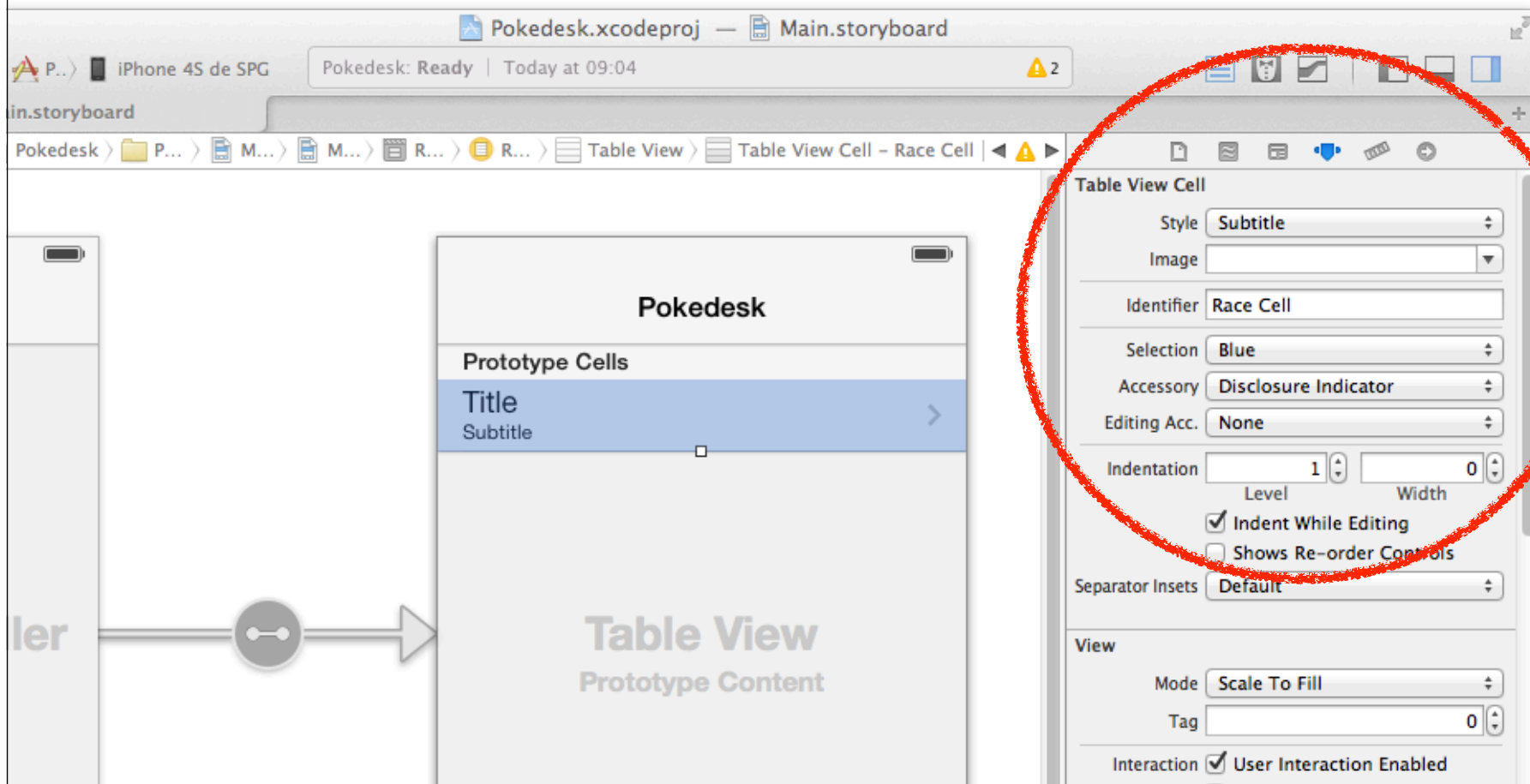


- Embeber la TableVC en un Navigation Controller.
  - Seleccionar en el IB el objeto RacesTableViewController, y ejecutar:  
*Menú Editor > Embed In > Navigation Controller*
- Poner *Pokedesk* como título en las barra de navegación.
  - Haciendo doble-click en la barra de navegación y tecleando el título.

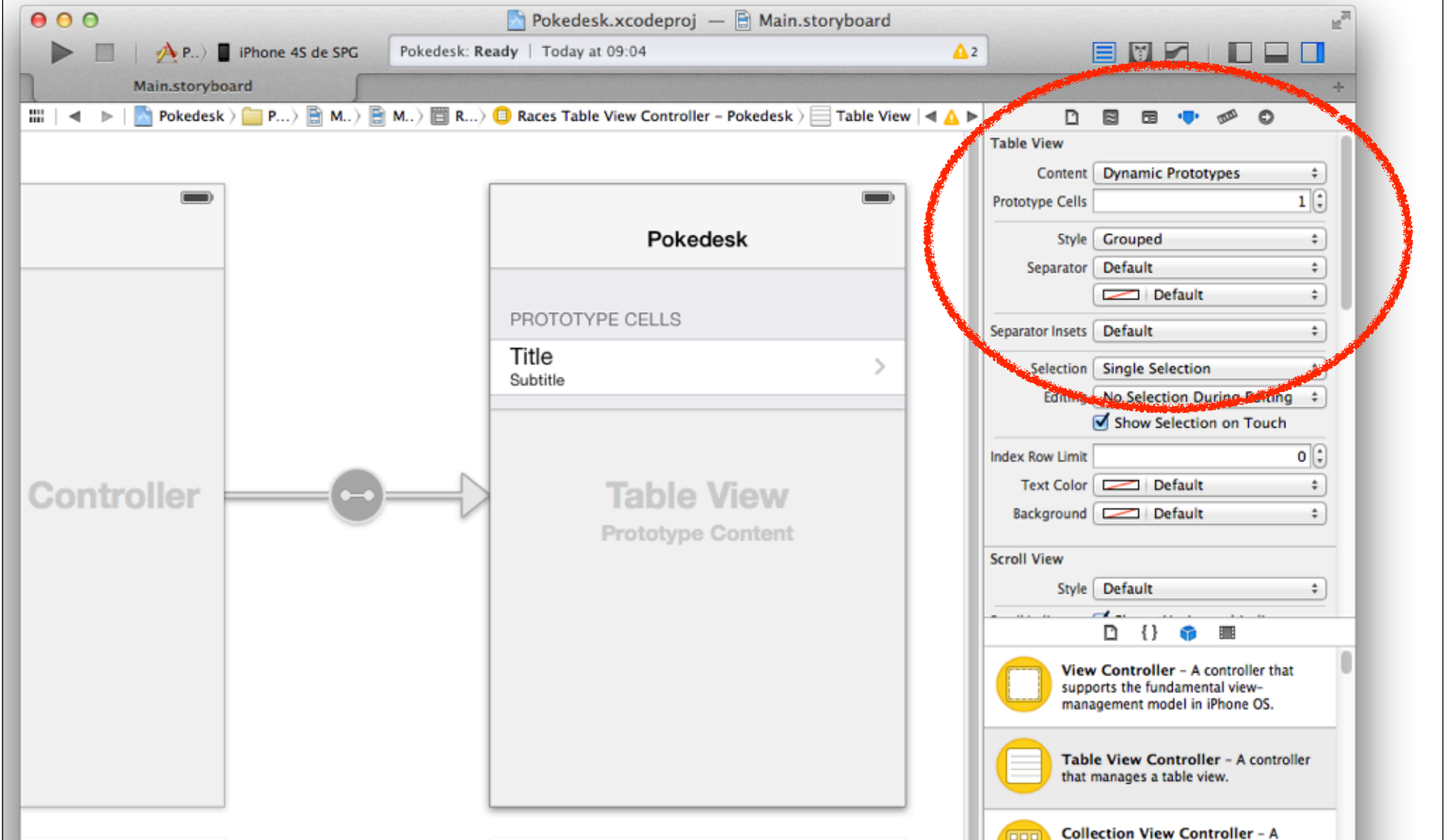




- Las celdas de la TableView de RacesTableVC se crearán usando un prototipo de celda.
- Editar propiedades del prototipo de celda:
  - Style: **Subtitle**
  - Identificador: **Race Cell**
  - Accessory: **Disclosure Indicator**



- Editar propiedades de la UITableView:
  - Style: **Grouped**



# El Modelo (MVC)

- Los datos que usará la aplicación están almacenados en un fichero de propiedades (plist) que nos proporcionan:
  - **pokemons.plist**
    - Contiene los nombres de las razas de los pokemons, sus códigos, los nombres de los ficheros de iconos, los tipos de razas, y las razas de cada tipo.
- También nos dan hecha una clase **PokedeskModel** que carga los datos de pokemons.plist y los hace públicos a través de dos propiedades:
  - **races**: un array con las razas.
  - **types**: un array con los tipos.
- También disponemos de dos clases:
  - **Race** - representar una raza de los pokemon.
    - Tiene propiedades para el nombre , el código, y nombre del fichero del icono de la raza.
  - **Type** - representar un tipo de pokemon.
    - Tiene propiedades para el nombre del tipo, y para referenciar a todas las razas del tipo.
- Los ficheros **pokemons.plist**, **PokedeskModel.h**, **PokedeskModel.m**, **Race.h**, **Race.m**, **Type.h**, **Type.m** y todos los ficheros con los iconos están disponibles para su descarga en la página de Apuntes de iOS.

# pokemons.plist

Key	Type	Value
▼ Root	Dictionary	(3 items)
▶ iconos	Dictionary	(299 items)
▶ nombres	Dictionary	(251 items)
▶ tipos	Dictionary	(18 items)

Key	Type	Value
▼ Root	Dictionary	(3 items)
▼ iconos	Dictionary	(299 items)
1	String	001.gif
10	String	010.gif
100	String	100.gif
101	String	101.gif
102	String	102.gif

Key	Type	Value
▼ Root	Dictionary	(3 items)
▶ iconos	Dictionary	(299 items)
▼ nombres	Dictionary	(251 items)
1	String	Bulbasaur
10	String	Caterpie
100	String	Voltorb
101	String	Electrode
102	String	Exeggcute

Key	Type	Value
▼ Root	Dictionary	(3 items)
▶ iconos	Dictionary	(299 items)
▶ nombres	Dictionary	(251 items)
▼ tipos	Dictionary	(18 items)
▼ acero	Array	(4 items)
Item 0	Number	205
Item 1	Number	212
Item 2	Number	227
Item 3	Number	208
▶ agua	Array	(50 items)
▶ bicho	Array	(22 items)
▼ demonio	Array	(6 items)
Item 0	Number	229
Item 1	Number	228
Item 2	Number	198
Item 3	Number	215

```
/**
 * Modelo de datos de los pokemons.
 * Tiene informacion sobre las razas y los tipos de pokemons.
 */
@interface PokedeskModel : NSObject

/**
 * Array de objetos Race.
 */
@property (nonatomic, strong) NSArray* races;

/**
 * Array de objetos Type.
 */
@property (nonatomic, strong) NSArray* types;

@end
```

**PokedeskModel.h**

```

/**
 * Clase para representar una raza de pokemon.
 */
@interface Race : NSObject

@property (nonatomic, strong) NSNumber* code;
@property (nonatomic, strong) NSString* name;
@property (nonatomic, strong) NSString* icon;

/**
 * Inicializador de la clase Race.
 * @param code Código de la raza.
 * @param name Nombre de la raza.
 * @param icon Nombre del fichero del icono de la raza.
 */
- (instancetype) initWithCode:(NSNumber*)code
                        name:(NSString*)name
                        icon:(NSString*)icon;

@end

```

Race.h

```

/**
 * Clase para representar un tipo de pokemon.
 * A un tipo de pokemon pertenecen varias razas.
 */
@interface Type : NSObject

@property (nonatomic, strong) NSString* name;
@property (nonatomic, strong) NSArray* razes;

/**
 * Inicializador de la clase Type.
 * @param name Nombre del tipo.
 * @param razes Todas las razas de este tipo.
 * Es un array de objetos Race.
 */
- (instancetype) initWithName: (NSString*)name
                razes: (NSArray*)razes;

@end

```

Type.h

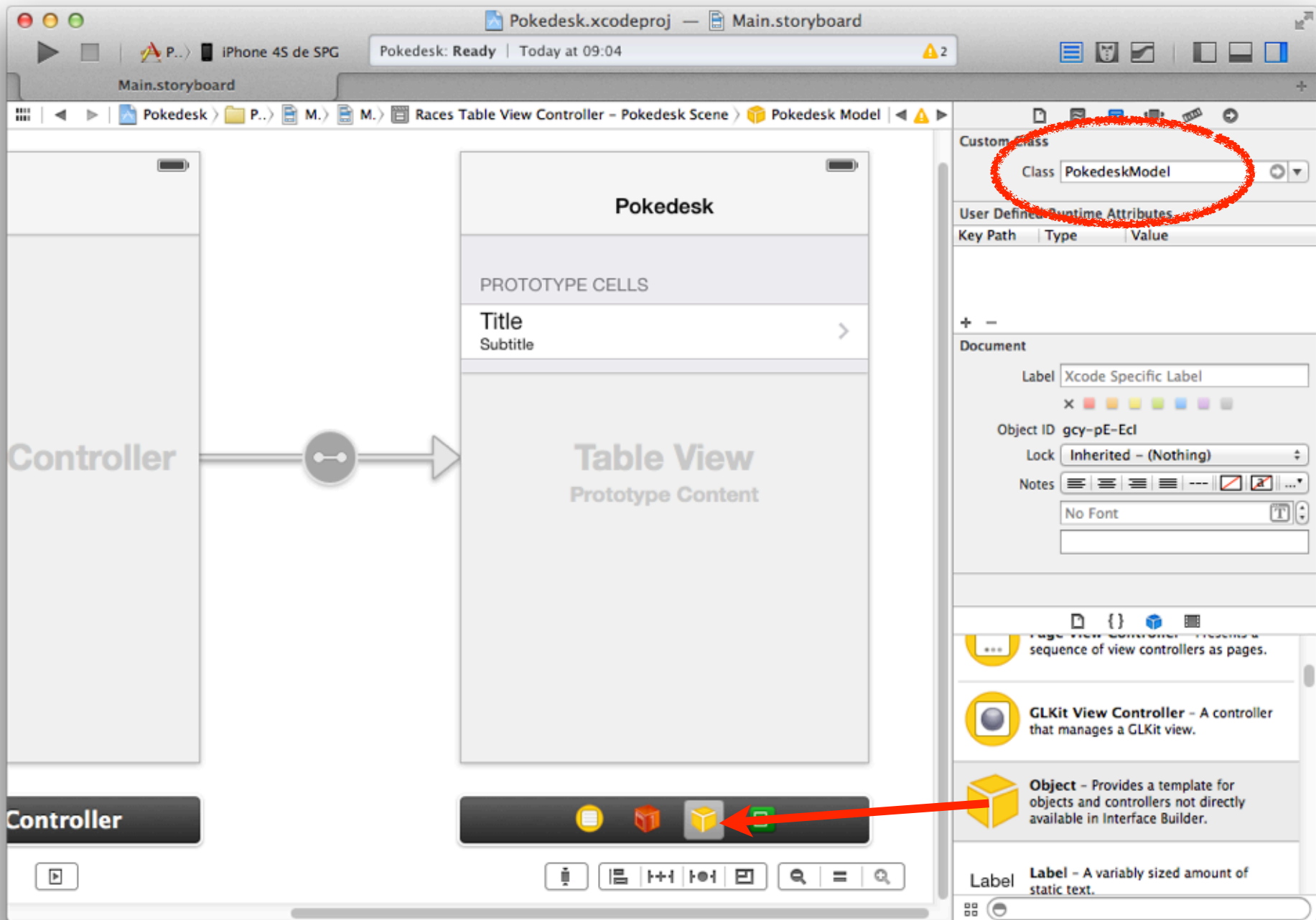
# Añadir al Proyecto

- Copiar al proyecto:
  - el directorio con los iconos
  - el fichero de propiedades pokemons.plist
  - los ficheros PokedeskModel.h y .m
  - los ficheros Race.h y .m
  - los ficheros Type.h y .m



# Crear el Modelo

- Crear un objeto para el modelo de la tabla (la M del MVC):
  - El modelo es un objeto de la clase **PokedeskModel**.
  - Para crear este objeto:
    - Con Interface Builder, arrastrar un objeto **NSObject** desde la librería de objetos hasta la escena **RacesTableViewController**.
    - Cambiar la clase del objeto arrastrado de **NSObject** a **PokedeskModel**.
    - Crear una propiedad **outlet** llamada **pokedeskModel** en **RacesTableViewController.m** que apunte al objeto **PokedeskModel**.
  - Alternativa: También se podría haber creado la propiedad **pokedeskModel** y el objeto **PokedeskModel** programáticamente.



**Ctrl+BI**

```
// RacesTableViewController.m
// Pokedesk
//
// Created by Santiago Pavón on 20/10/13.
// Copyright (c) 2013 UPM. All rights reserved.
//

#import "RacesTableViewController.h"
#import "PokedeskModel.h"

@interface RacesTableViewController ()

@property (strong, nonatomic) IBOutlet PokedeskModel *pokedeskModel;
@end

@implementation RacesTableViewController

- (id)initWithStyle:(UITableViewStyle)style
{
    self = [super initWithStyle:style];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    // Uncomment the following line to preserve selection between
    // presentations.
    // self.clearsSelectionOnViewWillAppear = NO;

    // Uncomment the following line to display an Edit button in the
    // navigation bar for this view controller.
    // self.navigationItem.rightBarButtonItem = self.editButtonItem;
}
```

# Data Source

- **RacesTableViewController** es el data source de su TableView interna.
- Implementaremos en **RacesTableViewController** los siguientes métodos del protocolo UITableViewDataSource:
  - **numberOfSectionsInTableView:**
  - **tableView:numberOfRowsInSection:**
  - **tableView:titleForHeaderInSection:**
  - **tableView:cellForRowAtIndexPath:**

```

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return [self.pokedeskModel.types count];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    Type * type = self.pokedeskModel.types[section];
    return [type.races count];
}

- (NSString *)      tableView:(UITableView *)tableView
    titleForHeaderInSection:(NSInteger)section
{
    Type * type = self.pokedeskModel.types[section];
    return type.name;
}

- (UITableViewCell *)tableView:(UITableView *)tableView
    cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Race Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier
                                                                forIndexPath:indexPath];

    Type * type = self.pokedeskModel.types[indexPath.section];
    Race * race = type.races[indexPath.row];

    // Rellenar la celda:
    cell.imageView.image = [UIImage imageNamed:race.icon];
    cell.textLabel.text = race.name;
    cell.detailTextLabel.text = [race.code description];

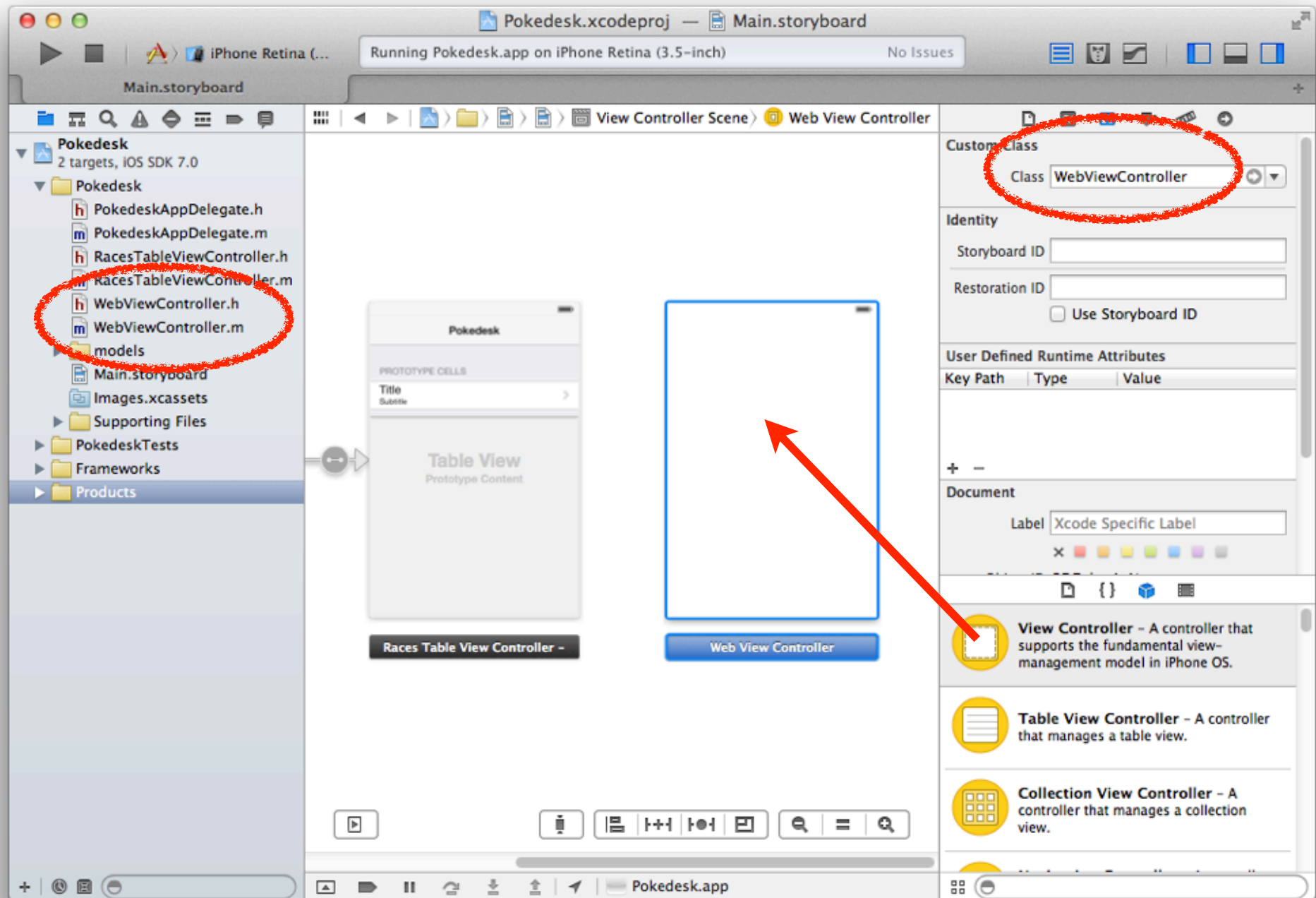
    return cell;
}

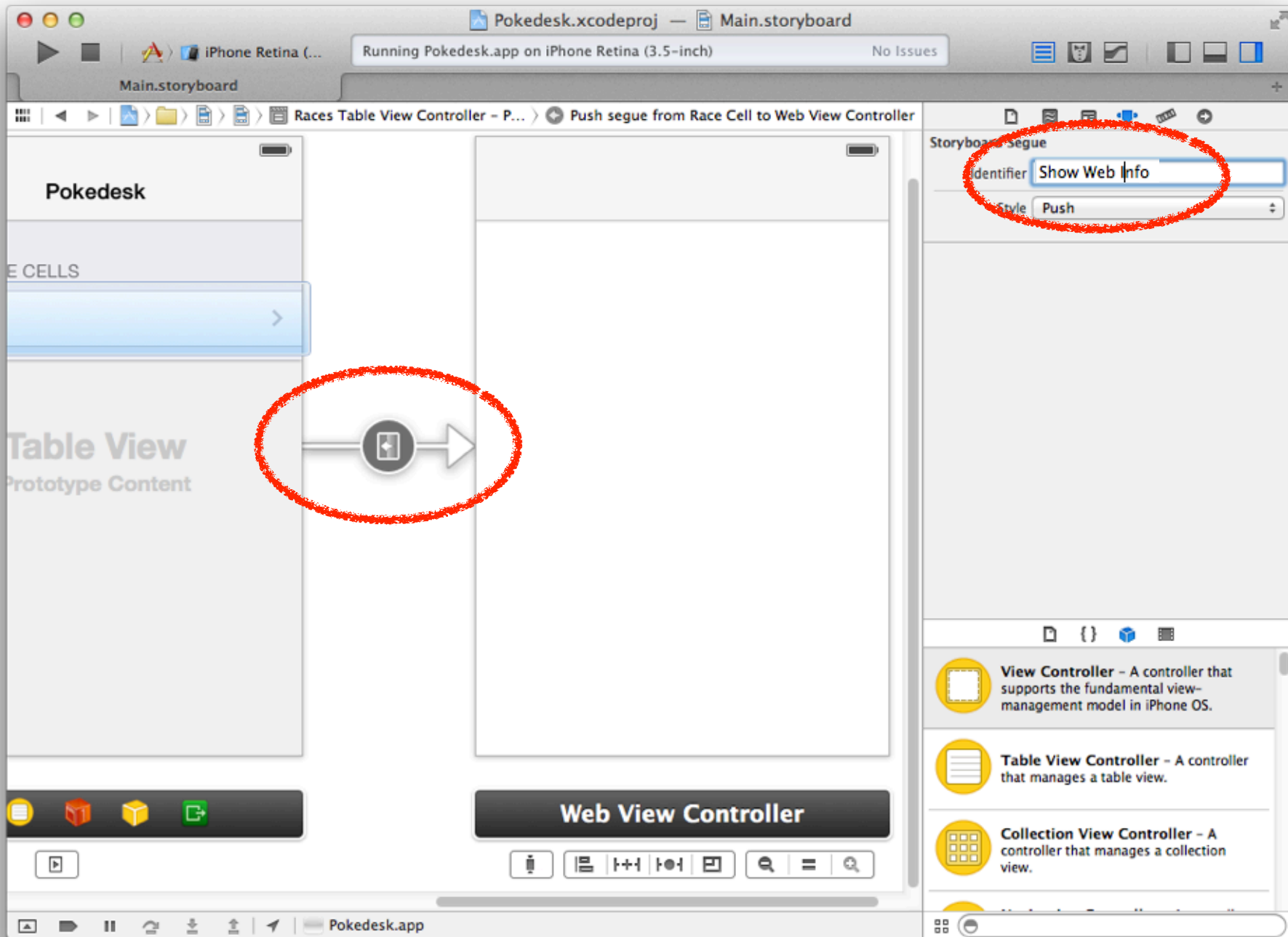
```

# Información Web

- Usaremos una segunda pantalla VC para mostrar la información web sobre la raza de pokemon que seleccionemos en la tabla.
- Usando Interface Builder, añadir al Storyboard un ViewController nuevo.
  - Crear una clase nueva para él.
    - Derivada de **UIViewController**.
    - Llamarla **WebViewController**.
    - Cambiar en el storyboard la clase del VC a **WebViewController**.
- Crear un **Segue** de tipo **Push** desde el prototipo de celda de la tabla de **RacesTableViewController** hasta **WebViewController**.
  - Se crea con **Ctrl+Arrastar** desde el prototipo de la celda hasta la escena **WebViewController**.
  - Poner como identificador de este segue el valor **Show Web Info**.
- La página web que mostraremos será:

[http://es.pokemon.wikia.com/wiki/NOMBRE\\_DE\\_LA\\_RAZA](http://es.pokemon.wikia.com/wiki/NOMBRE_DE_LA_RAZA)





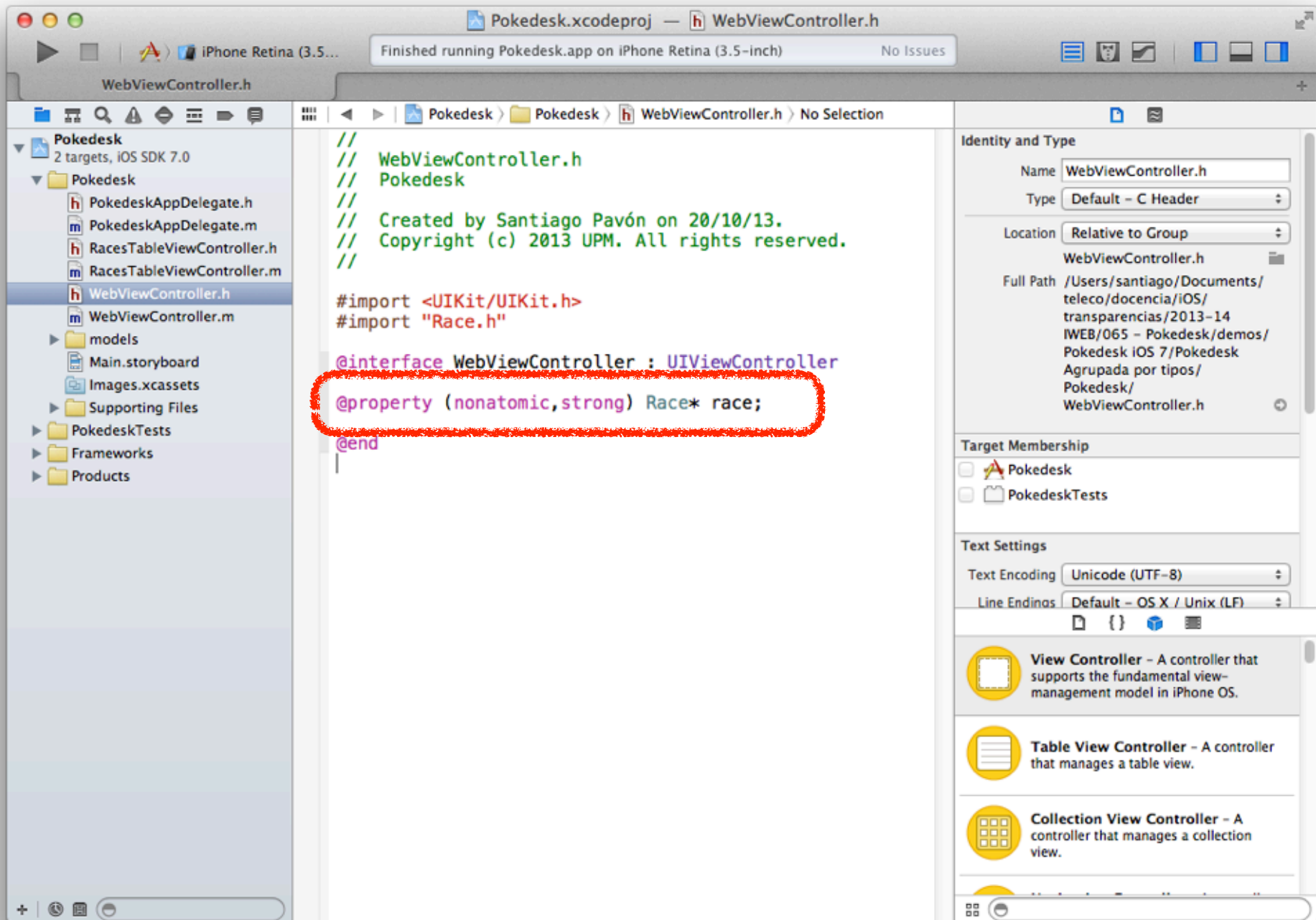


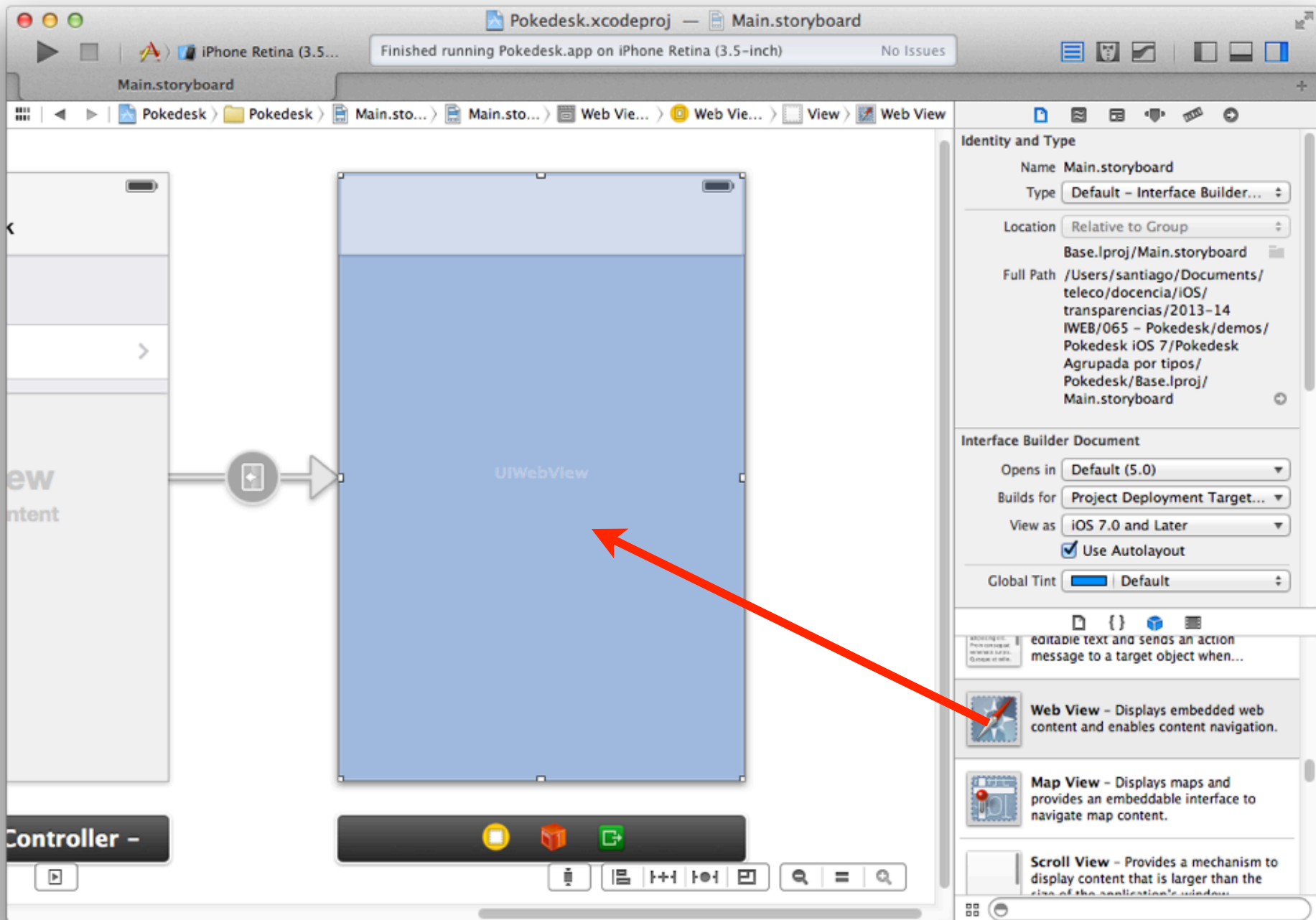
# WebViewController

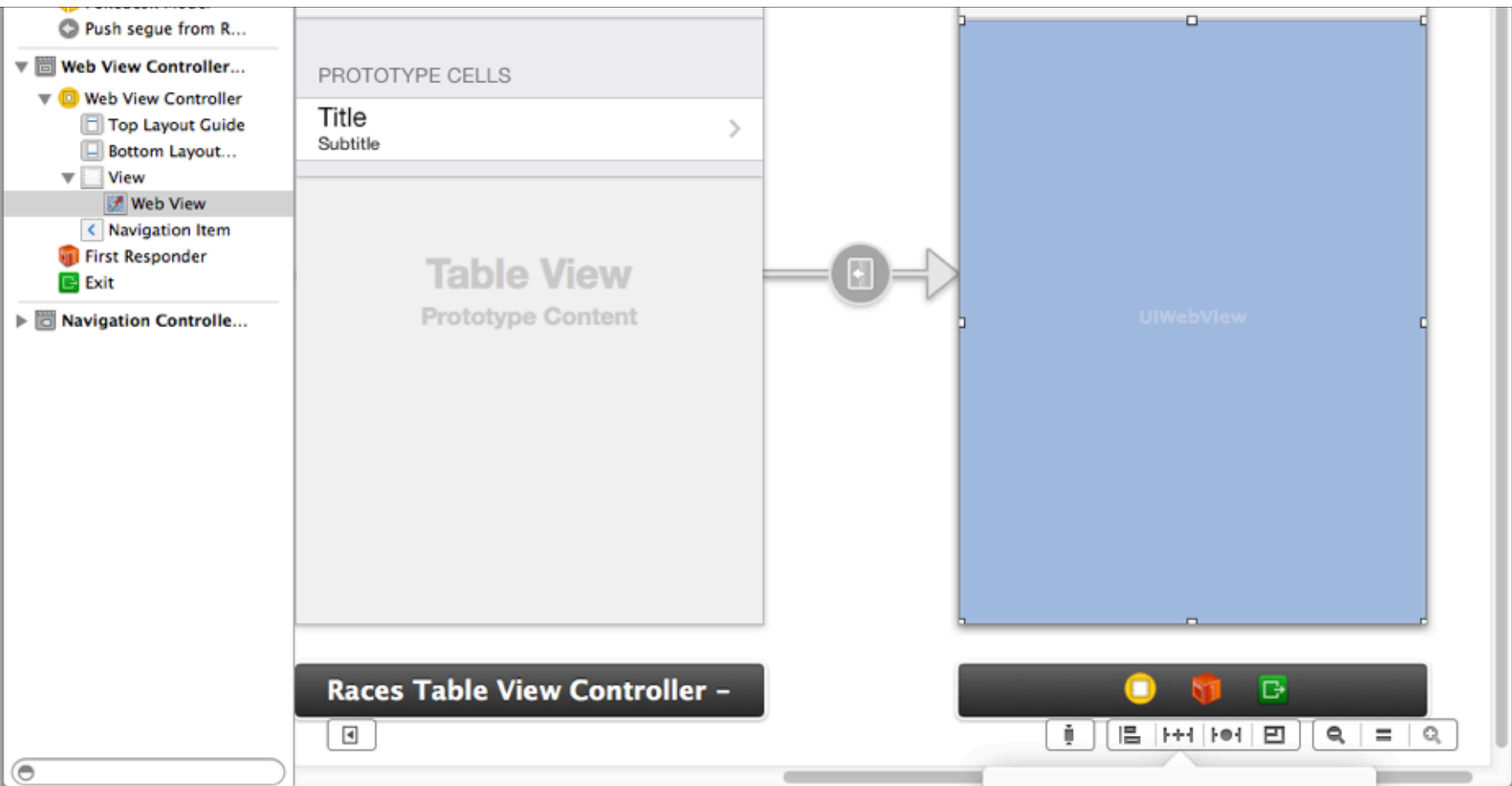
- Crear en **WebViewController.h** una propiedad llamada **race**.

```
@property (nonatomic, strong) Race* race;
```

- Es el parámetro de entrada de esta VC donde pasaremos la raza del pokemon a mostrar.
- Usando Interface Builder, editar la escena **WebViewController**.
  - Añadir un objeto **UIWebView**.
    - Añadir restricciones de AutoLayout para que ocupe todo el área bajo la barra de navegación.
  - Crear un **outlet** al objeto **UIWebView** añadido.
  - Añadir a la barra de navegación un **Bar Button Item** que al pulsarse invoque la acción **goBack** de la **UIWebView**.
    - Configurar el botón para que muestre el texto que se desee.







**Add New Constraints**

0

q 0

0

Spacing to nearest neighbor

Width 320

Height 414

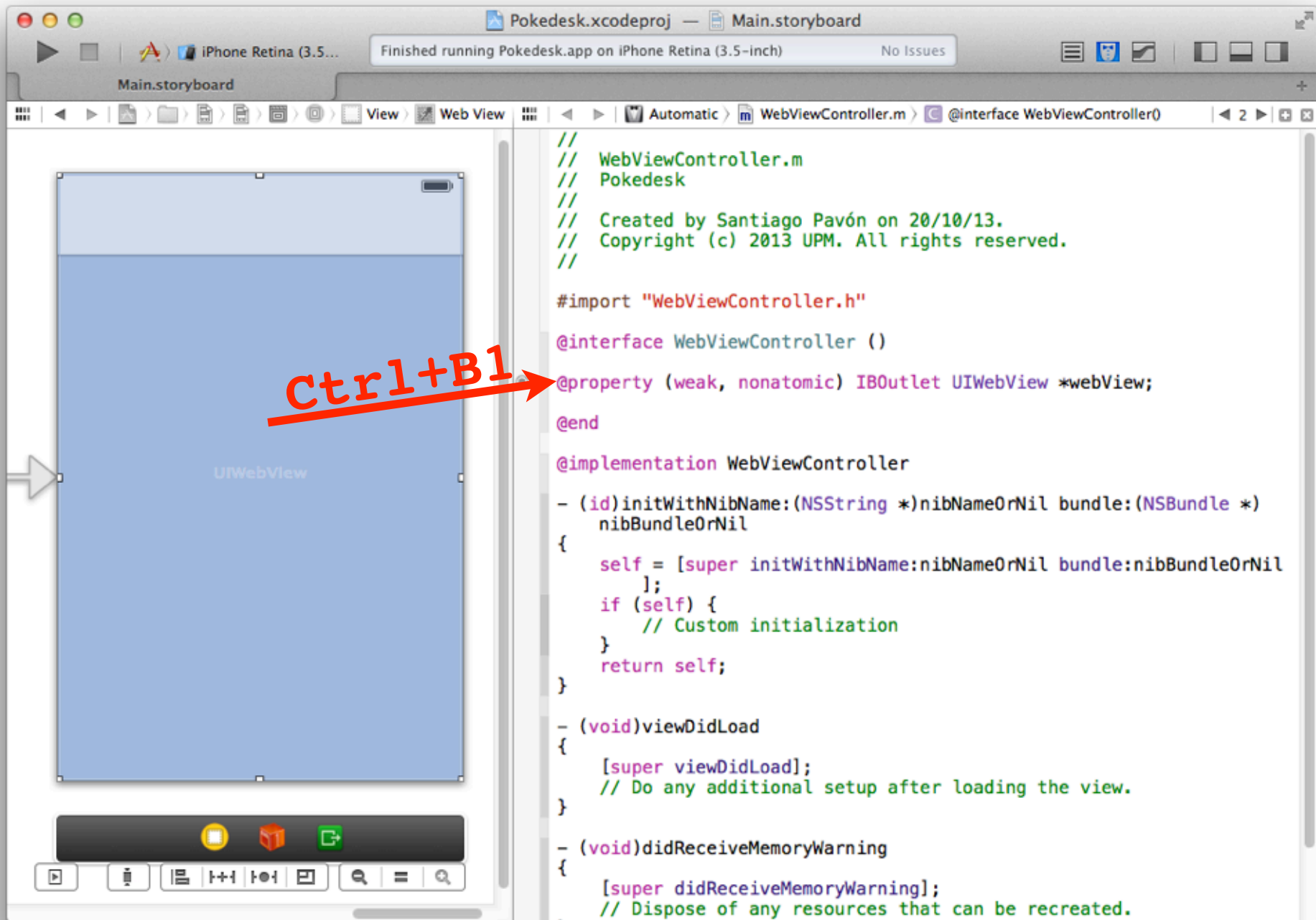
Equal Widths

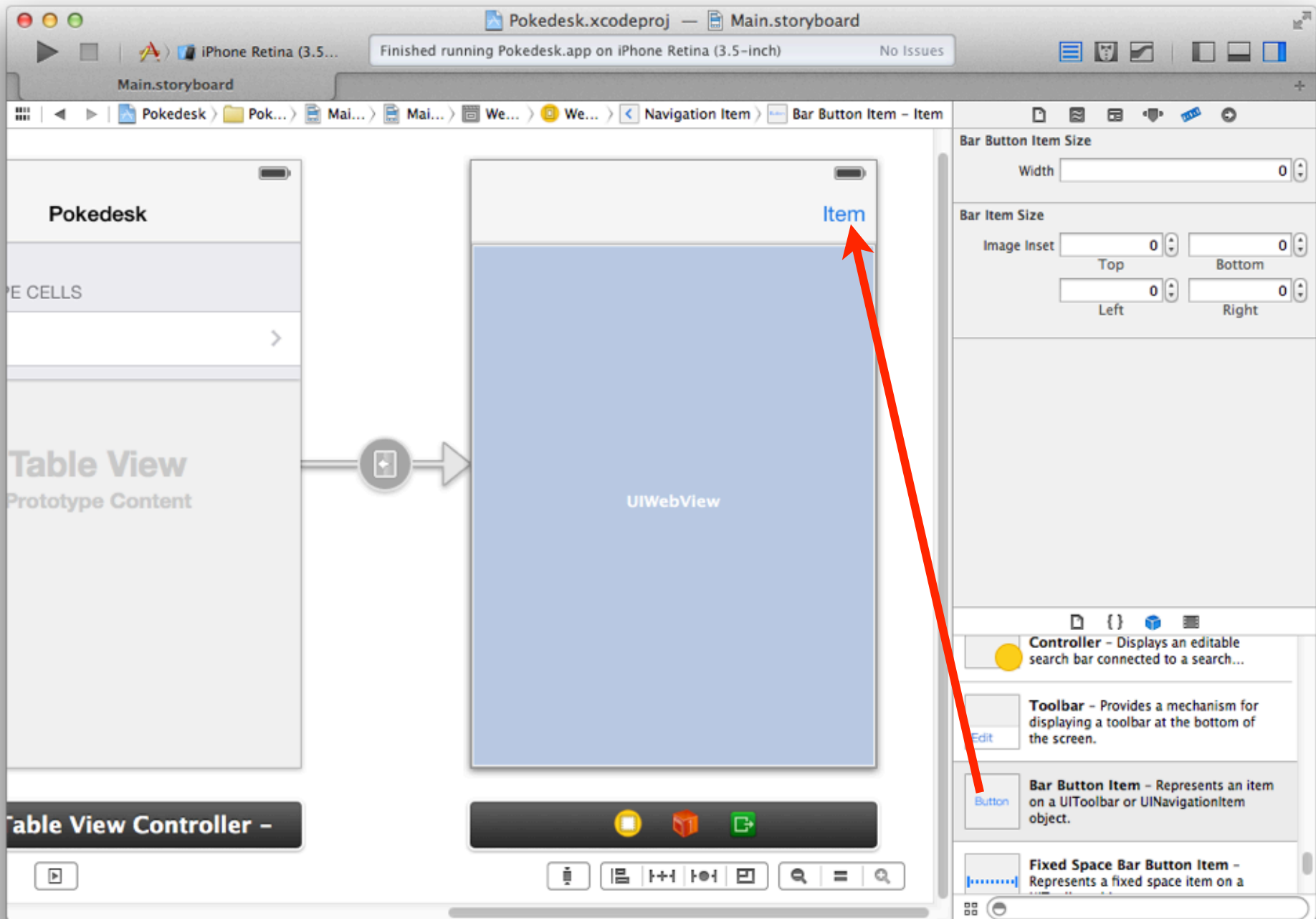
Equal Heights

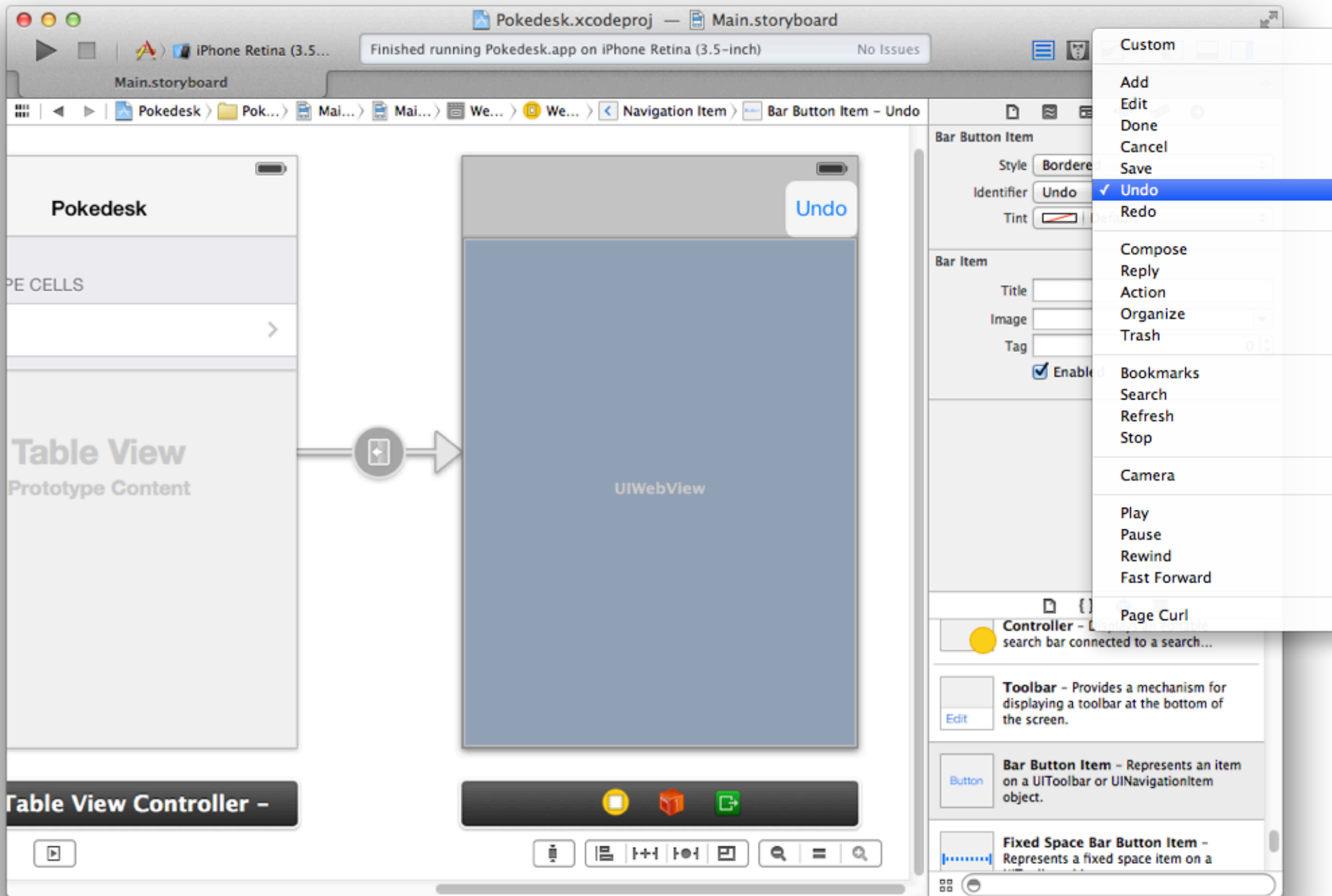
Align Leading Edges

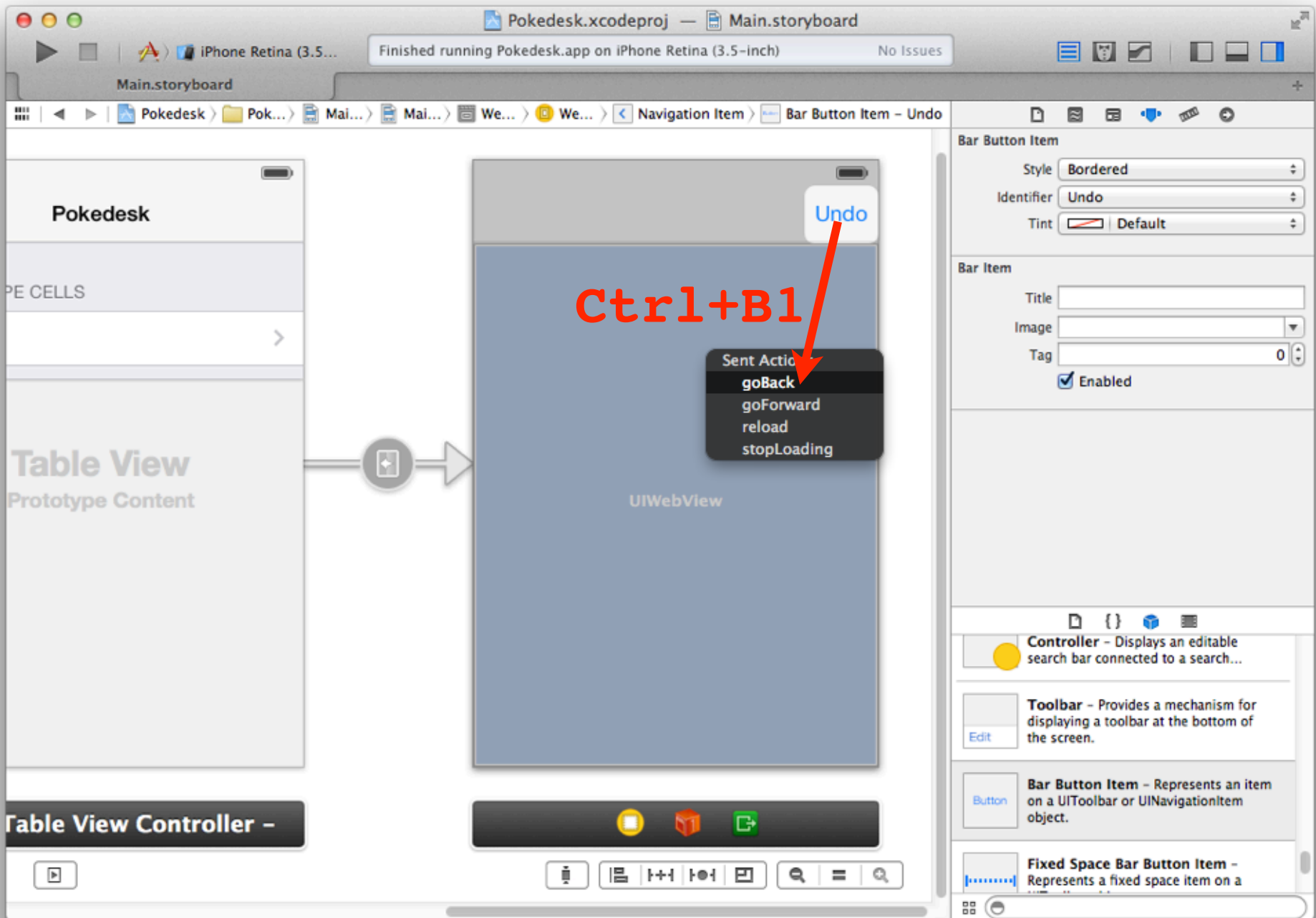
Update Frames None

Add 4 Constraints

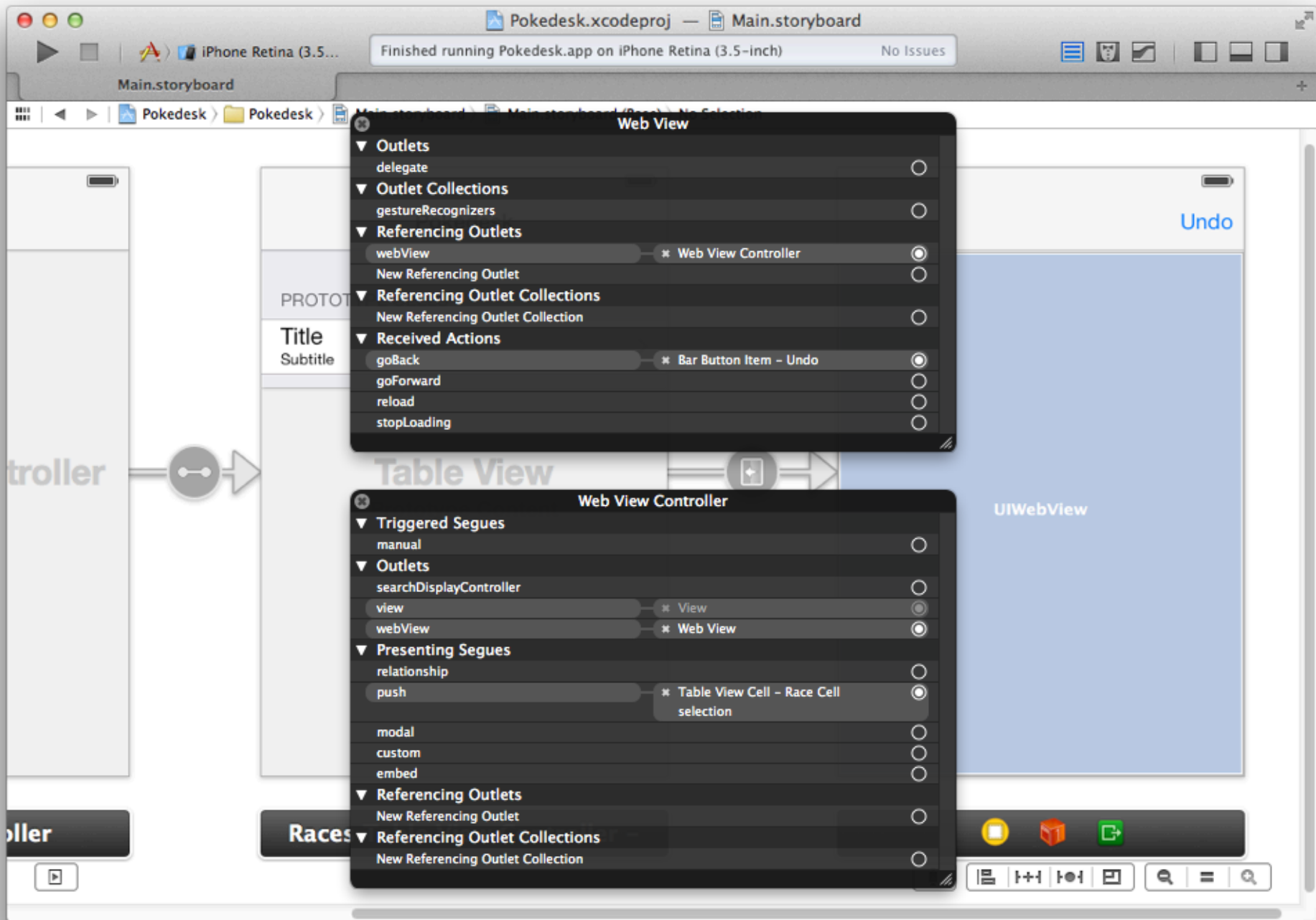












- En el método **viewDidLoad** de **WebViewController**:
  - Poner el nombre de la raza como título de la barra de navegación.
  - Construir la URL a mostrar usando el valor guardado en **self.race.name**, y presentar la URL en la **WebView**.

```

#import "WebViewController.h"

@interface WebViewController ()
@property (weak, nonatomic) IBOutlet UIWebView *webView;
@end

@implementation WebViewController

- (void)viewDidLoad
{
    [super viewDidLoad];

    // Poner nombre de la pokemon como titulo de la Navigation Bar
    self.title = self.race.name;

    // Cargar la URL del pokemon
    NSString *str = [NSString stringWithFormat:
        @"http://es.pokemon.wikia.com/wiki/%@", self.race.name];

    // Escapar caracteres conflictivos
    str = [str stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];

    NSURL *url = [NSURL URLWithString:str];
    NSURLRequest *req = [NSURLRequest requestWithURL:url];
    [self.webView loadRequest:req];
}

@end

```

WebViewController.m

# Preparar Segue

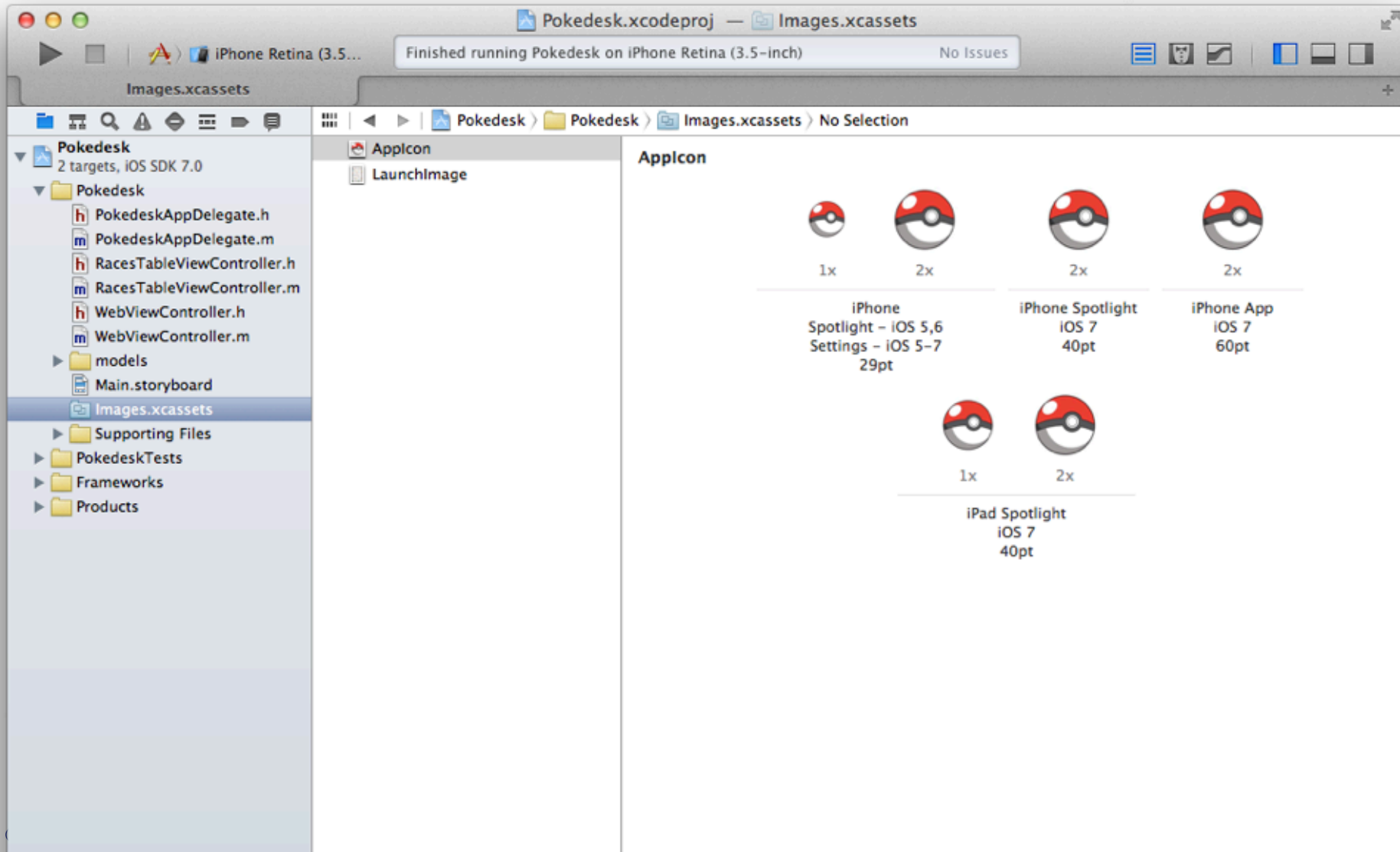
- **En RacesTableViewController:**

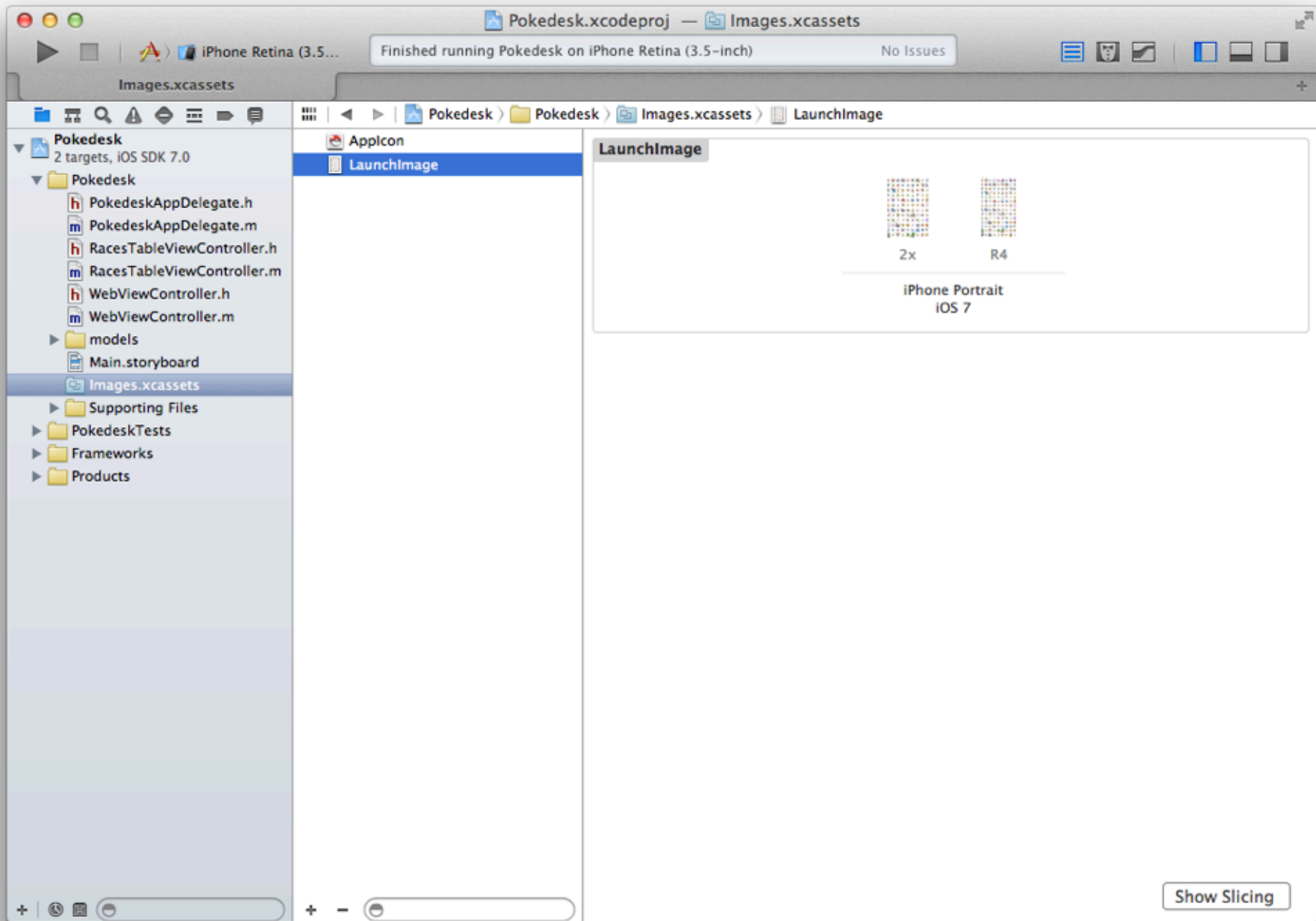
```
- (void)prepareForSegue:(UIStoryboardSegue *)segue
    sender:(id)sender
{
    if ([segue.identifier isEqualToString:@"Show Web Info"]) {
        NSIndexPath * ip = [self.tableView indexPathForCell:sender];

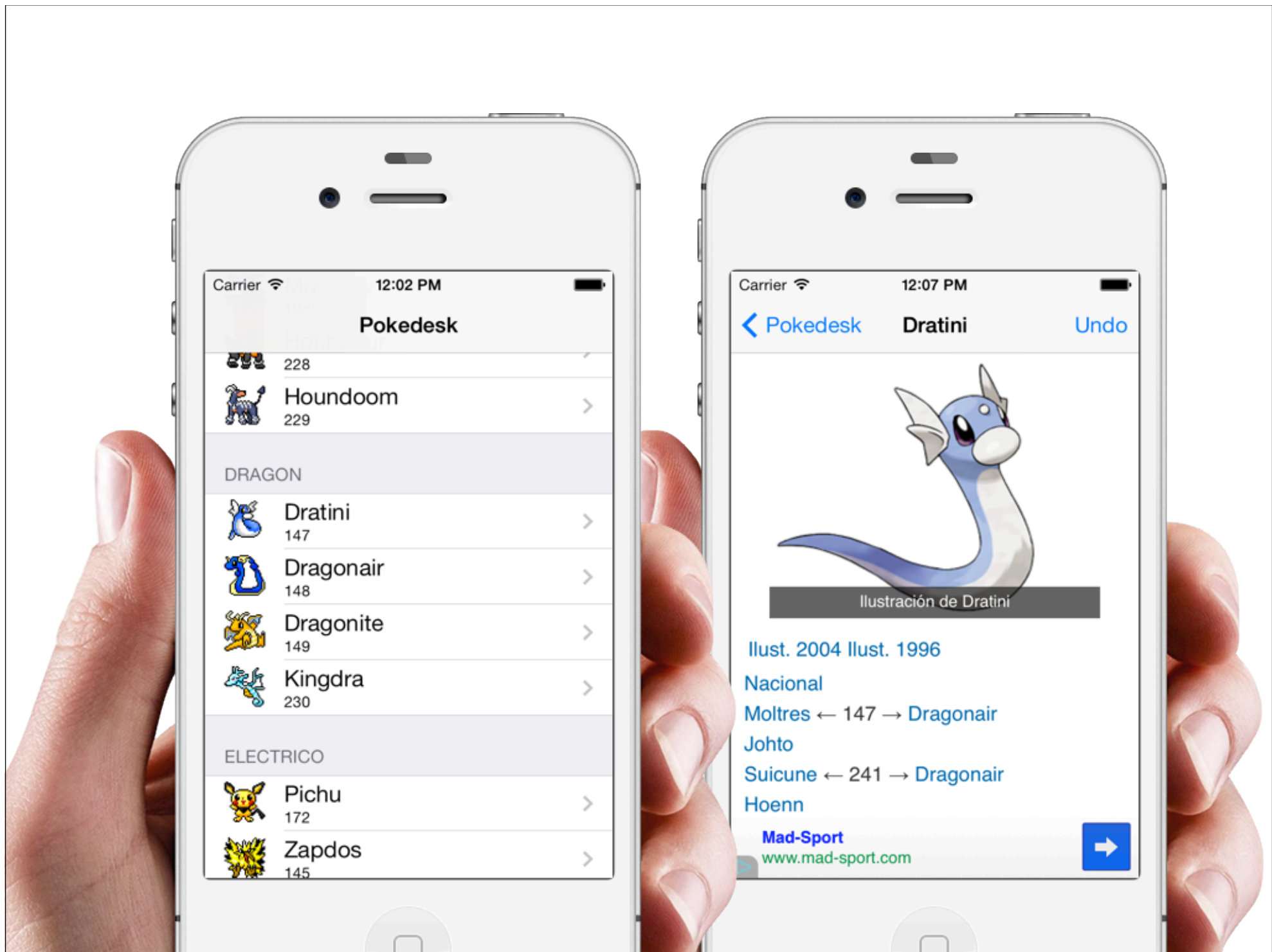
        WebViewController *wvc = segue.destinationViewController;

        Type * type = self.pokedeskModel.types[ip.section];
        wvc.race = type.races[ip.row];
    }
}
```

# Añadir Iconos e Imágenes Iniciales







# Ejercicios

- Que se muestre una tabla sólo con los tipos
  - Y al pulsar sobre un tipo se muestran las razas de ese tipo en otra pantalla.
- Meter una barra de búsqueda.
- Crear una app universal.
  - En un iPad mostraría un Split View Controller.



