



POLITÉCNICA

ETSIT
UPM

dit
UPM

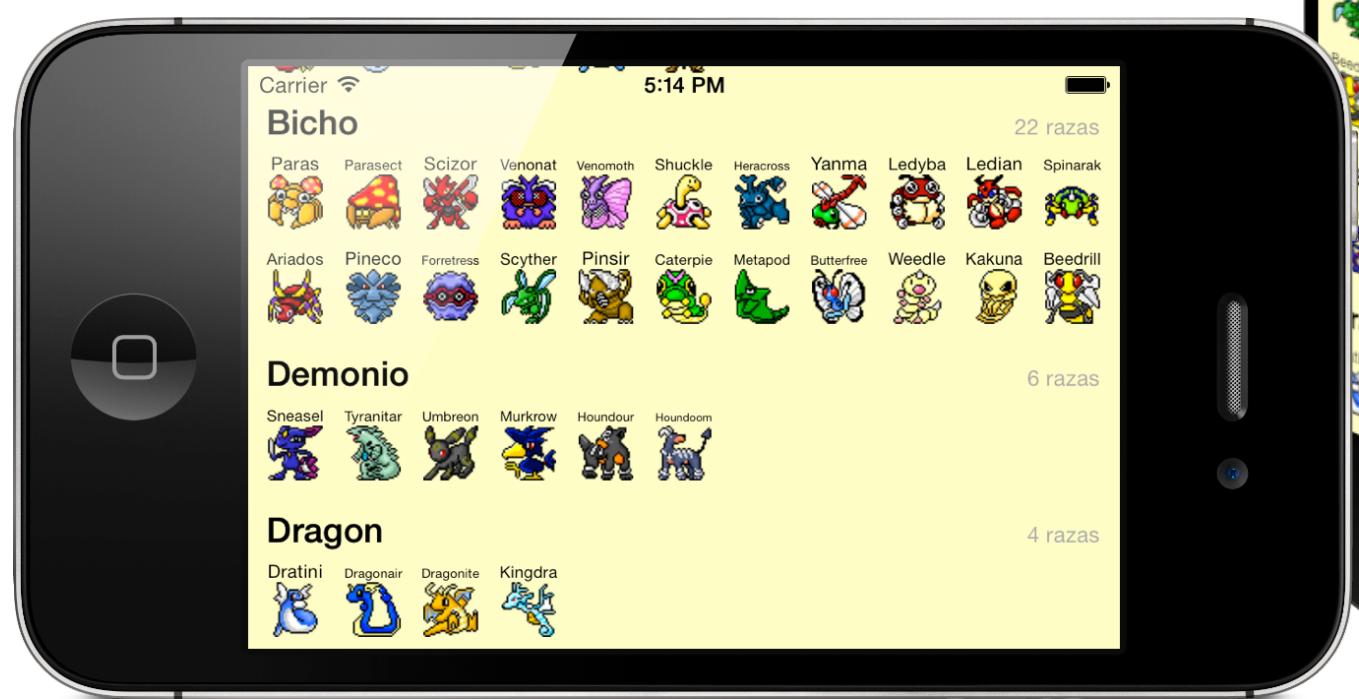
Desarrollo de Apps para iOS Collection Views

IWEB,LSWC 2013-2014
Santiago Pavón

ver: 2014.03.29

Características de las CollectionViews

- Muestran los datos usando celdas dispuestas con diferentes layouts.
 - Actualmente solo se soporta Flow Layout.
 - Coloca las celdas en una parrilla completando líneas horizontal o verticalmente.
- Las celdas se puede agrupar en secciones.
 - Las secciones pueden tener un vistas complementarias
 - haciendo el papel de cabecera y pie de las secciones.
- El contenido de las celdas es siempre personalizado.
- Realiza un uso eficiente de celdas y vistas complementarias.
 - Reutilizando las no visibles.



Clases y Protocolos

- Algunas clases:
 - **UICollectionView**: la vista que contiene las celdas.
 - **UICollectionViewCell**: la celda con los datos.
 - **UICollectionViewController**: un UIViewController contenido una UICollectionView ocupando todo el espacio.
 - **UICollectionViewFlowLayout**: clase del gestor de layout Flow.
 - **UICollectionViewLayoutAttributes**: clase con los atributos que aplican a las celdas o vistas complementarias.
- Algunos protocolos:
 - **UICollectionViewDataSource**: define el protocolo de obtención de datos del objeto dataSource.
 - **UICollectionViewDelegate**: define el protocolo para comunicarse con el delegado informando de sucesos.

UICollectionView

- Propiedades:

dataSource

delegate

backgroundView

collectionViewLayout

allowsSelection

allowsMultipleSelection

- Algunos métodos:
 - `initWithFrame:collectionViewLayout:`
 - `registerClass:forCellReuseIdentifier:`
 - `registerNib:forCellReuseIdentifier:`
 - `registerClass:forSupplementaryViewOfKind:withReuseIdentifier:`
 - `registerNib:forSupplementaryViewOfKind:withReuseIdentifier:`
 - `dequeueReusableCellWithIdentifier:forIndexPath:`
 - `dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath:`
 - `setCollectionViewLayout:animated:completion:`
 - `reloadData`
 - `reloadSections:`
 - `reloadItemsAtIndexPaths:`
 - `numberOfSections`
 - `numberOfItemsInSection:`
 - `visibleCells`
 - `insertItemsAtIndexPaths:`
 - `moveItemAtIndexPath:toIndexPath:`
 - `deleteItemsAtIndexPaths:`
 - `indexPathsForSelectedItems`
 - `selectItemAtIndexPath:animated:scrollPosition:`
 - `deselectItemAtIndexPath:animated:`
 - `indexPathForItemAtPoint:`
 - `indexPathForCell:`
 - `cellForItemAtIndexPath:`
 - `scrollToItemAtIndexPath:atScrollPosition:animated:`
 - etc . . .

UICollectionViewCell

- Propiedades:

`contentView`

`backgroundView`

`selectedBackgroundView`

`selected`

`highlighted`

UICollectionViewDataSource

```
- (NSInteger)collectionView:(UICollectionView*)  
    numberOfRowsInSection:(NSInteger)  
  
- (NSInteger)numberOfSectionsInCollectionView:(UICollectionView*)  
  
- (UICollectionViewCell*)collectionView:(UICollectionView*)  
    cellForItemAtIndexPath:(NSIndexPath*)  
  
- (UICollectionReusableView*)collectionView:(UICollectionView*)  
    viewForSupplementaryElementOfKind:(NSString*)  
    atIndexPath:(NSIndexPath*)
```

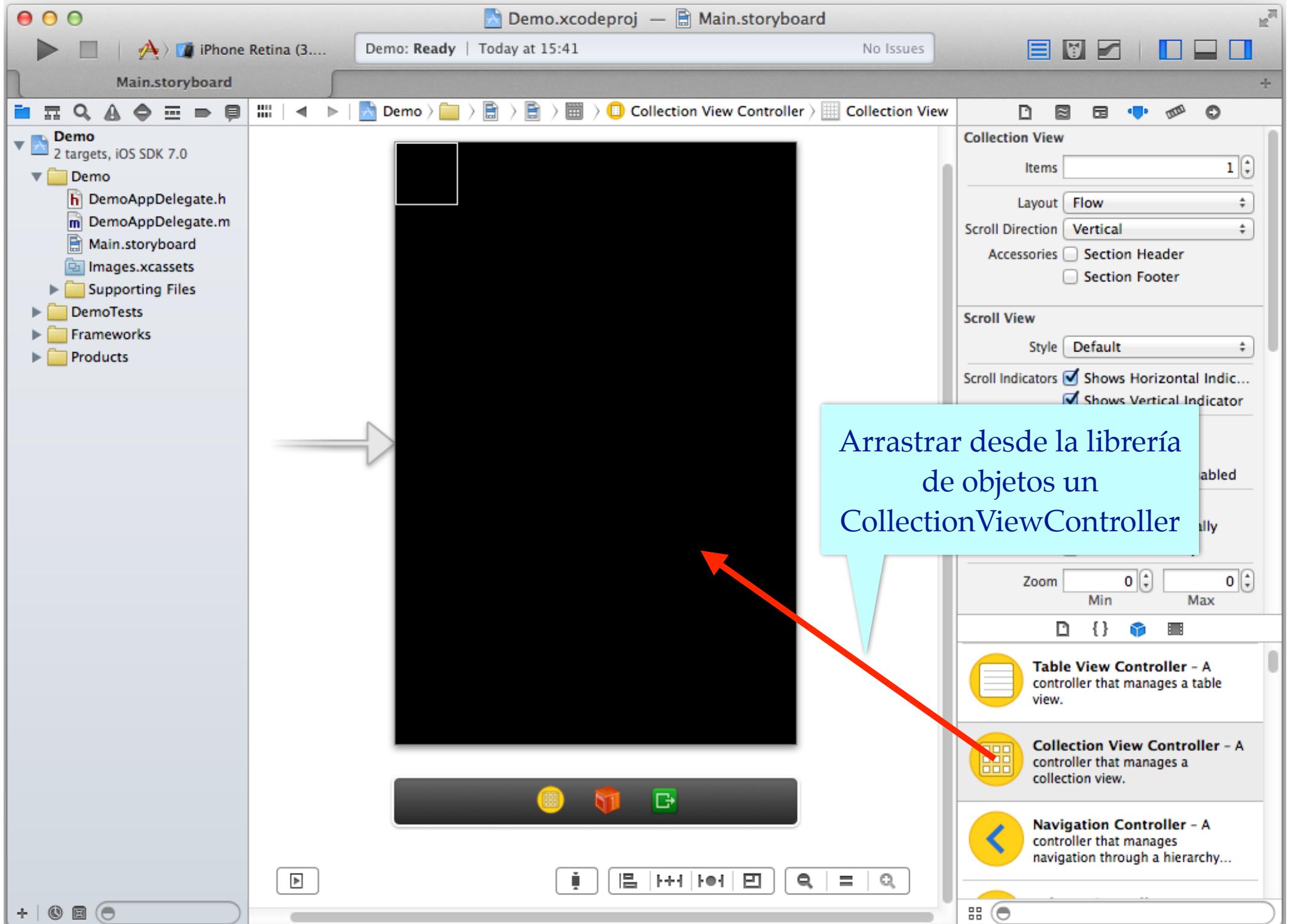
UICollectionviewDelegate

- Selección de celdas:
 - `collectionView:shouldSelectItemAtIndexPath:`
 - `collectionView:didSelectItemAtIndexPath:`
 - `collectionView:shouldDeselectItemAtIndexPath:`
 - `collectionView:didDeselectItemAtIndexPath:`
- Destacar celdas:
 - `collectionView:shouldHighlightItemAtIndexPath:`
 - `collectionView:didHighlightItemAtIndexPath:`
 - `collectionView:didUnhighlightItemAtIndexPath:`
- Eliminar vistas:
 - `collectionView:didEndDisplayingCell:forItemAtIndexPath:`
 - `collectionView:didEndDisplayingSupplementaryView:forElementOfKind:atIndexPath:`
- Transición entre layouts:
 - `collectionView:transitionLayoutForOldLayout:newLayout:`
- Gestión de acciones
 - `collectionView:shouldShowMenuForItemAtIndexPath:`
 - `collectionView:canPerformAction:forItemAtIndexPath:withSender:`
 - `collectionView:performAction:forItemAtIndexPath:withSender:`

UICollectionViewController

- Inicializador:
-initWithCollectionViewLayout:
- Propiedades:
collectionView
collectionViewLayout
clearsSelectionOnViewWillAppear
useLayoutForLayoutNavigationTransitions

Demo



Main.storyboard

Demo
2 targets, iOS SDK 7.0

- Demo
 - DemoAppDelegate
 - DemoAppDelegate
 - Main.storyboard
 - Images.xcasset
- Supporting File
- DemoTests
- Frameworks
- Products

Choose options for your new file:

Crear una clase nueva para la nueva pantalla.
Derivada de UICollectionViewController.

Class PokemonsViewController

Subclass of UICollectionViewController

- Targeted for iPad
- With XIB for user interface

Cancel

Previous

Next



Collection View Controller — A controller that manages a collection view.



Navigation Controller — A controller that manages navigation through a hierarchy...

Demo.xcodeproj — Main.storyboard

Demo: Ready | Today at 15:57 No Issues

Main.storyboard

Demo
2 targets, iOS SDK 7.0

Demo
DemoAppDelegate.h
DemoAppDelegate.m
PokemonsViewController.h
PokemonsViewController.m
Main.storyboard
Images.xcassets
Supporting Files
DemoTests
Frameworks
Products

Cambiar la clase del UICollectionViewController a PokemonsViewController.

Custom Class
Class PokemonsViewController

Identity
Storyboard ID
Restoration ID
 Use Storyboard ID

User Defined Runtime Attributes
Key Path | Type | Value

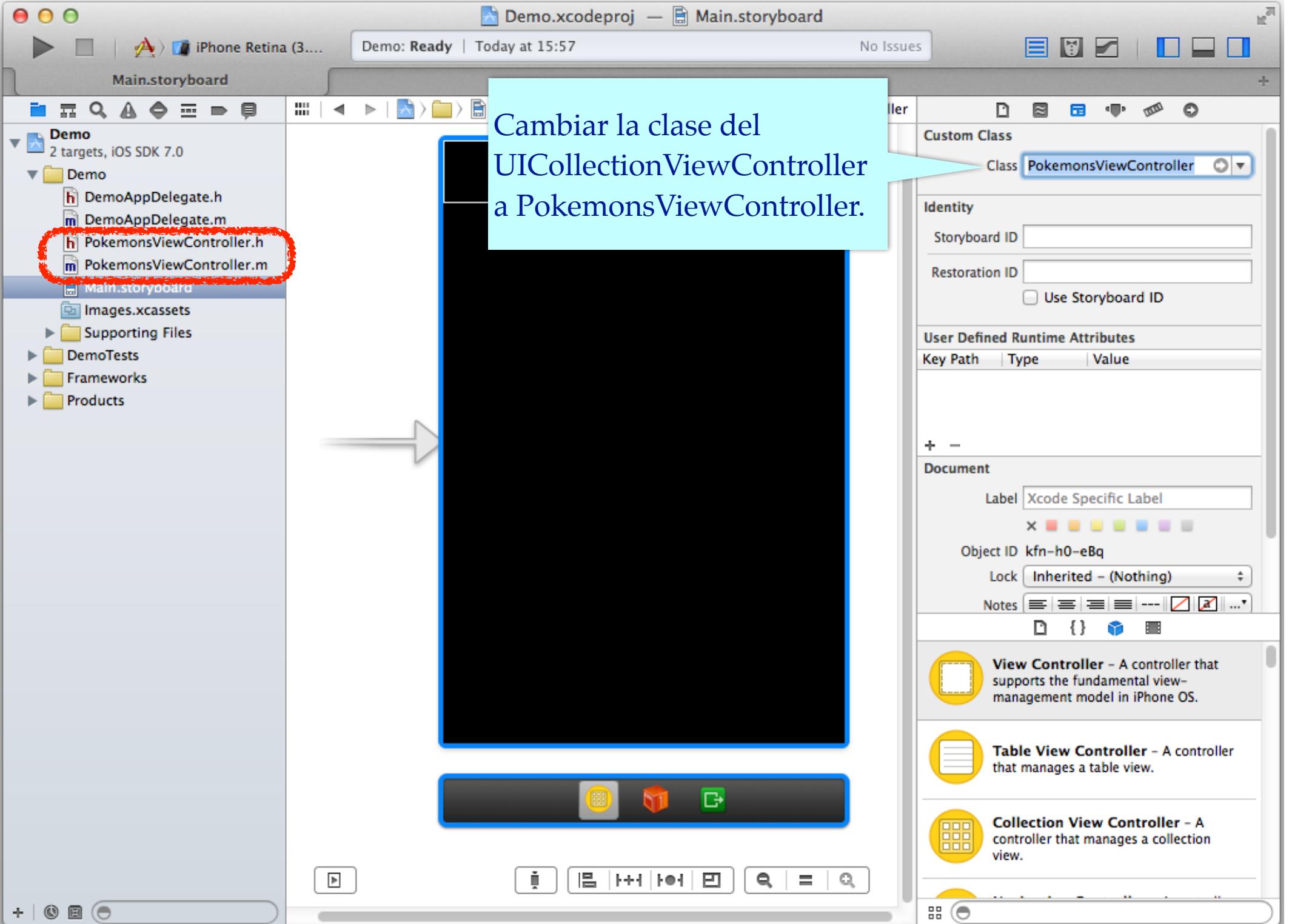
+ -

Document
Label Xcode Specific Label
Object ID kfn-h0-eBq
Lock Inherited – (Nothing)
Notes

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Collection View Controller – A controller that manages a collection view.



Demo.xcodeproj — Main.storyboard

Demo: Ready | Today at 15:57 No Issues

Main.storyboard

Demo 2 targets, iOS SDK 7.0

Demo

- DemoAppDelegate.h
- DemoAppDelegate.m
- PokemonsViewController.h
- PokemonsViewController.m

models

- PokedeskModel.h
- PokedeskModel.m
- pokemons.plist
- Race.h
- Race.m
- Type.h
- Type.m

icon

- 001.gif
- 002.gif
- 003.gif
- 004.gif
- 005.gif
- 006.gif
- 007.gif
- 008.gif
- 009.gif
- 010.gif
- 011.gif
- 012.gif
- 013.gif
- 014.gif
- 015.gif
- 016.gif
- 017.gif
- 018.gif
- 019.gif

Pokemons View Contr... Collection View

Collection View

- Items 1
- Layout Flow
- Scroll Direction Vertical
- Accessories Section Header
 Section Footer

ScrollView

- Style Default
- ScrollIndicator Shows Horizontal Indicator
 Shows Vertical Indicator
- Scrolling Scrolling Enabled
 Paging Enabled
 Direction Lock Enabled
- Bounce Bounces
 Bounce Horizontally
 Bounce Vertically
- Zoom 0 0

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Collection View Controller – A controller that manages a collection view.

Copiar al proyecto los ficheros del modelo de la Pokedex y los iconos.

Demo.xcodeproj — PokemonsViewController.m

Demo: Ready | Today at 15:57 No Issues

PokemonsViewController.m

```
// PokemonsViewController.m
// Demo
//
// Created by Santiago Pavón on 28/10/13.
// Copyright (c) 2013 UPM. All rights reserved.
//

#import "PokemonsViewController.h"

#import "PokedeskModel.h"
#import "Race.h"
#import "Type.h"

@interface PokemonsViewController : UIViewController

@property (nonatomic, strong) PokedeskModel * pokedeskModel;

@end

@implementation PokemonsViewController

- (id)initWithNibName:(NSString *)NibNameOrNil bundle:(NSBundle *)bundleOrNil
{
    ...
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.

    self.pokedeskModel = [[PokedeskModel alloc] init];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end
```

Esta propiedad es el modelo.

Demo.xcodeproj — Main.storyboard

Demo: Ready | Today at 15:57 No Issues

Main.storyboard

Pokemons View Controller Scene

Pokemons View Controller

Collection View

Collection View Cell - PokemonCell

Identifier: PokemonCell

Mode: Center

Tag: 0

Interaction: User Interaction Enabled, Multiple Touch

Alpha: 1

Background: Yellow

Tint: Default

Drawing: Opaque, Hidden, Clears Graphics Context, Clip Subviews, Autoresize Subviews

Stretching: X: 0, Y: 0, Width: 1, Height: 1

Collection Reusable View - Defines the attributes and behavior of reusable views in a collection view, such as a...

Image View - Displays a single image, or an animation described by an array of images.

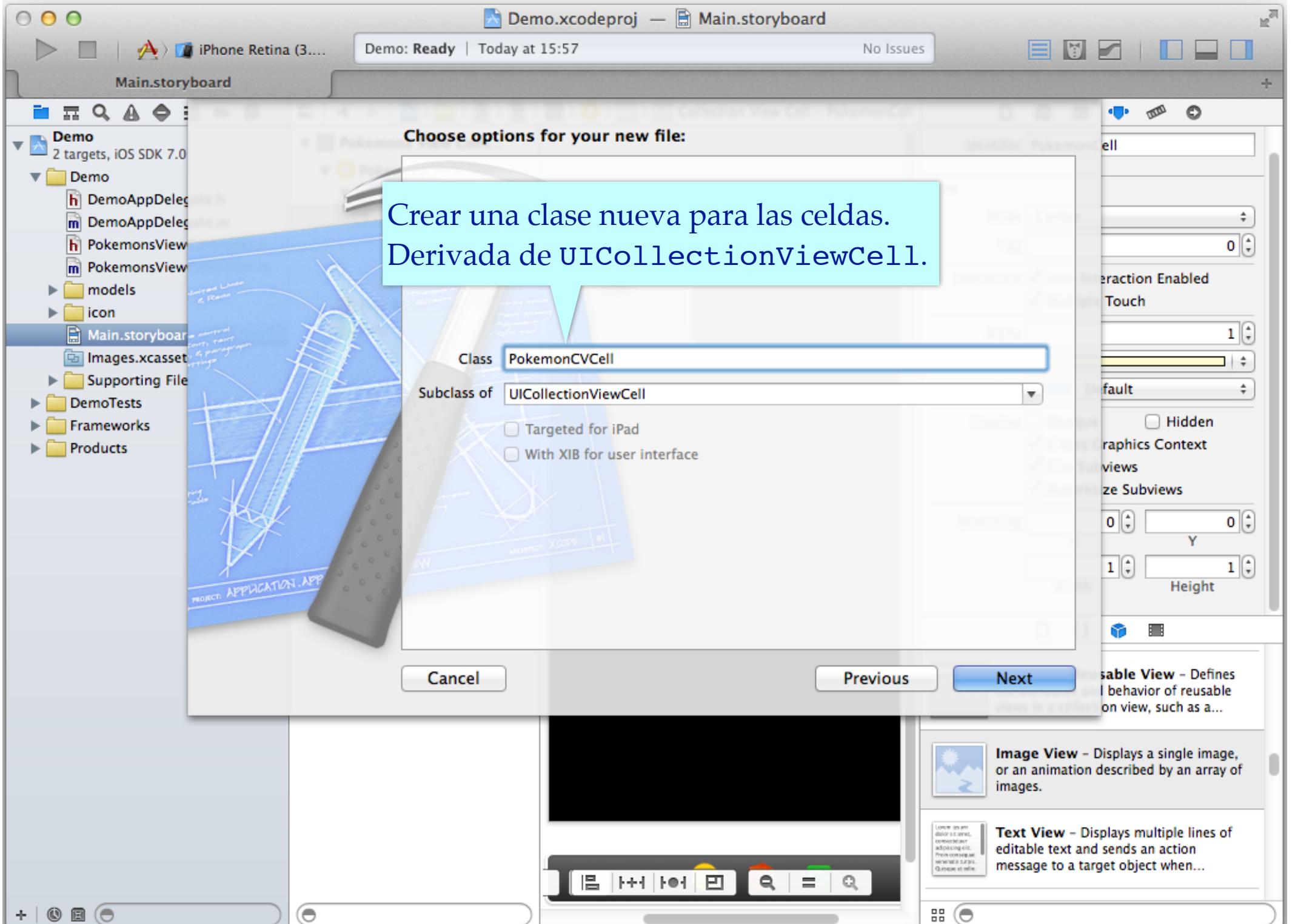
Text View - Displays multiple lines of editable text and sends an action message to a target object when...

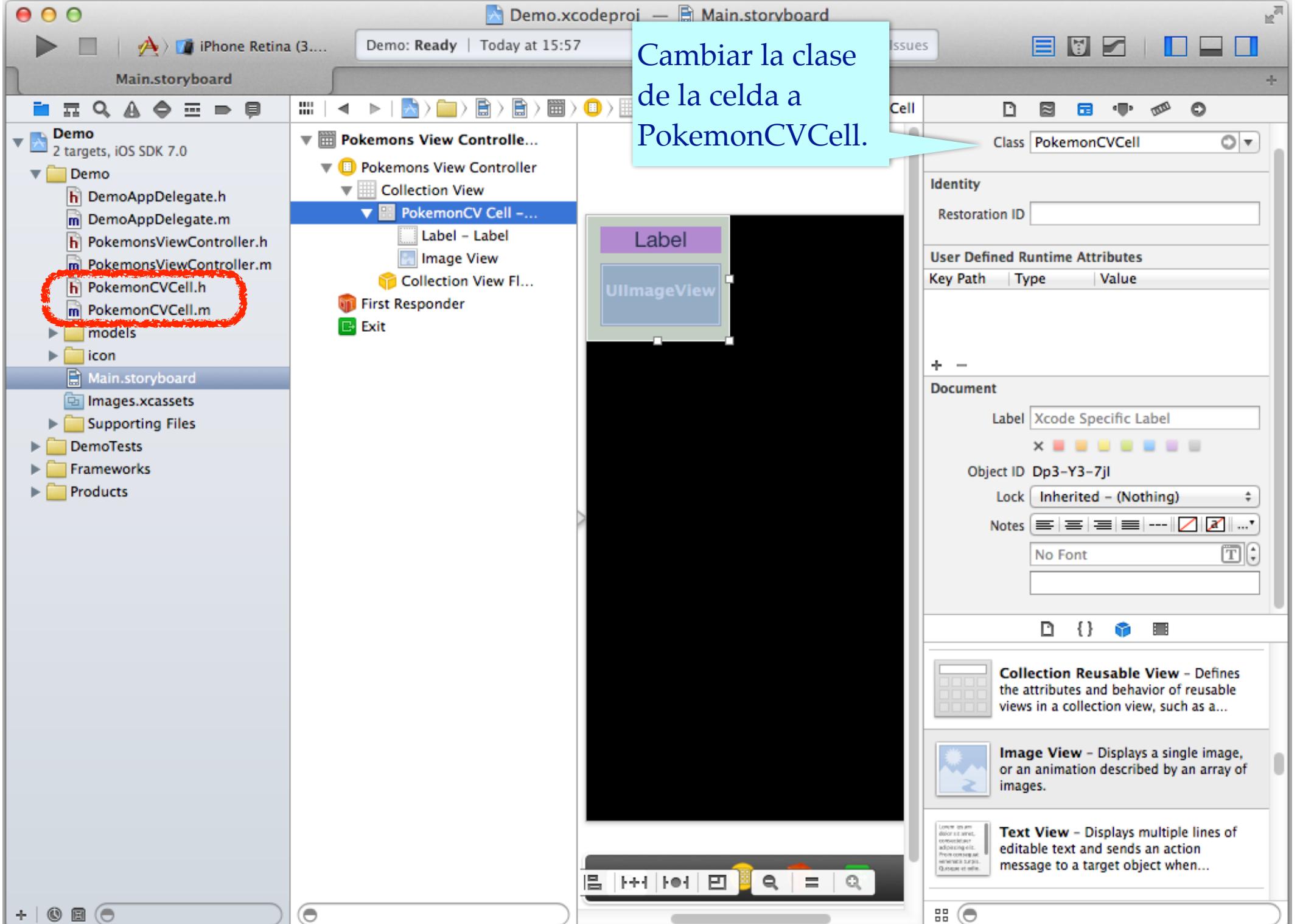
Label

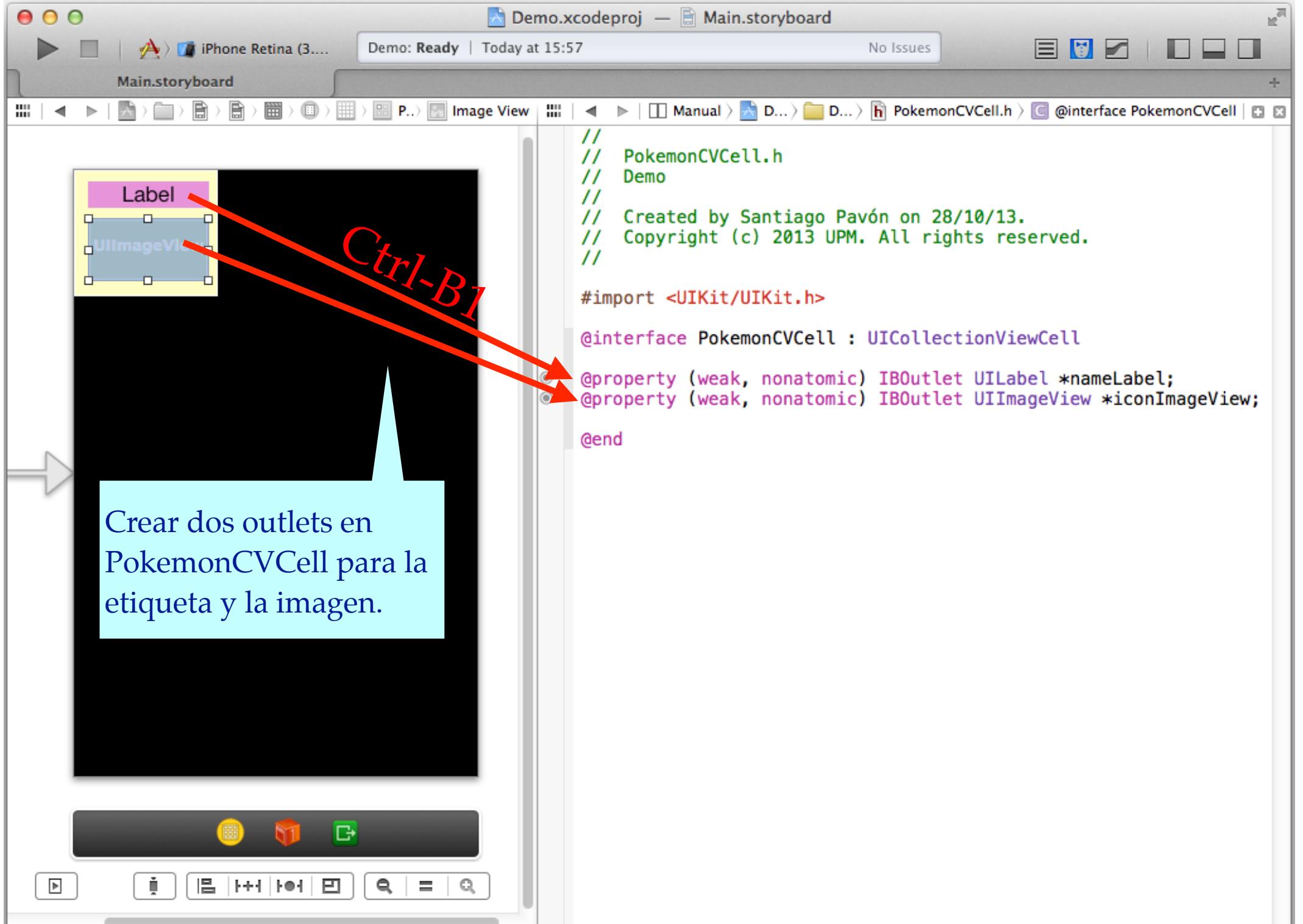
UIImageView

Poner un identificador de reutilización a la celda.

Diseñar la celda.
Añado una UILabel y una UIImageView desde la librería de objetos.







Demo.xcodeproj — PokemonsViewController.m

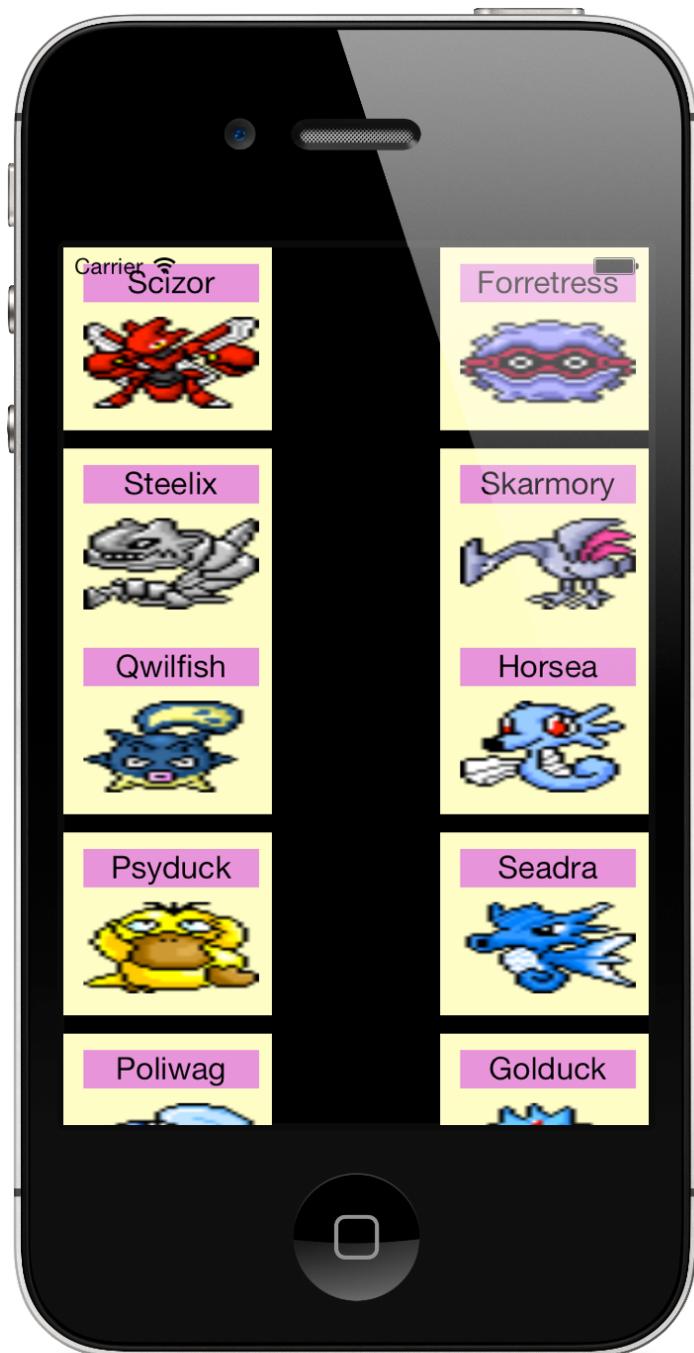
Finished running Demo on iPhone Retina (3.5-inch) No Issues

PokemonsViewController.m

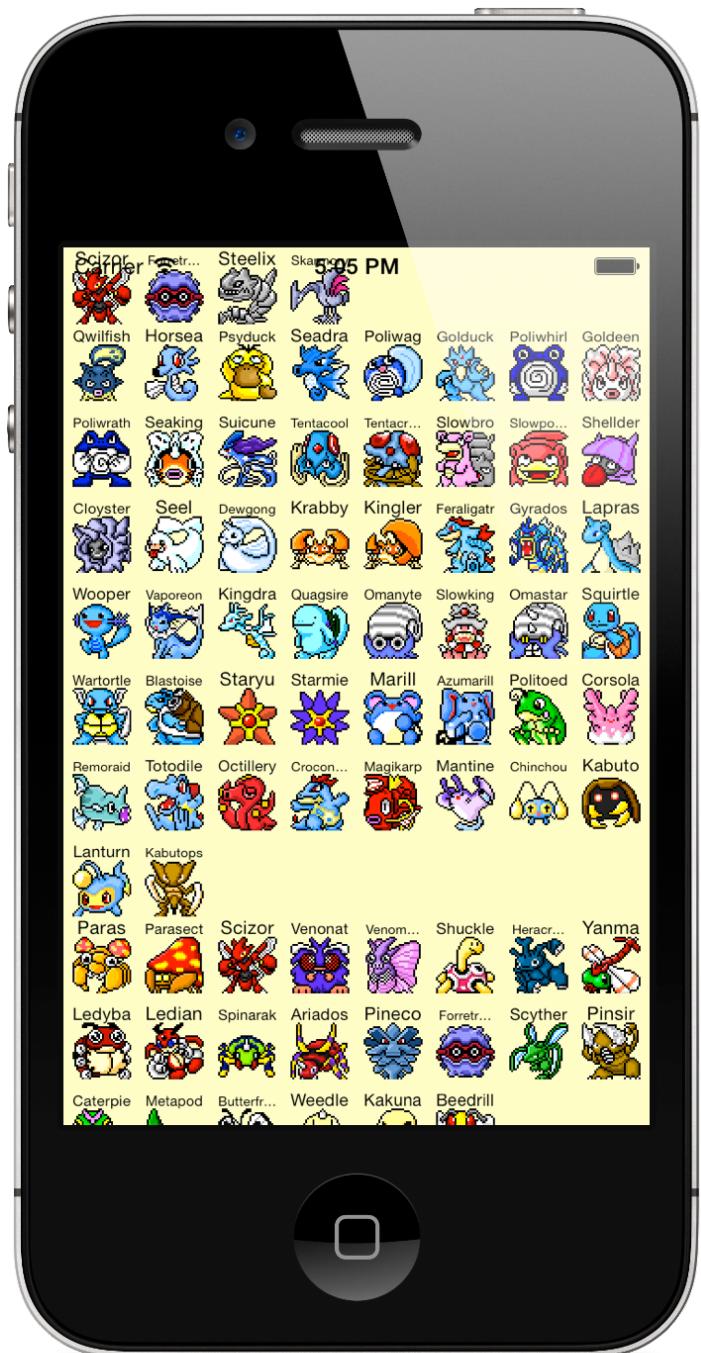
```
PokemonsViewController.m | Demo | Demo | PokemonsViewController.m | -collectionView:numberOfItemsInSection:  
// Dispose of any resources that can be recreated.  
}  
  
#pragma mark - Collection View Data Source  
  
- (NSInteger) numberOfSectionsInCollectionView:(UICollectionView *)collectionView  
{  
    return [self.pokedexModel.types count];  
}  
  
- (NSInteger) collectionView:(UICollectionView *)collectionView  
    numberOfItemsInSection:(NSInteger)section  
{  
    Type * type = self.pokedexModel.types[section];  
    return [type.races count];  
}  
  
- (UICollectionViewCell *)collectionView:(UICollectionView *)collectionView  
    cellForItemAtIndexPath:(NSIndexPath *)indexPath  
{  
    static NSString * cellId = @"PokemonCell";  
  
    PokemonCVCell * cell = [collectionView dequeueReusableCellWithReuseIdentifier:cellId  
        forIndexPath:indexPath];  
  
    NSInteger section = indexPath.section;  
    NSInteger item = indexPath.item;  
  
    Type * type = self.pokedexModel.types[section];  
    Race * race = type.races[item];  
  
    cell.nameLabel.text = race.name;  
    cell.iconImageView.image = [UIImage imageNamed:race.icon];  
  
    return cell;  
}
```

Métodos del Data Source

Añadir arriba:
#import "PokemonCVCell.h"



Retocar tamaños,
colores, fonts en el
inspector.



Demo.xcodeproj — Main.storyboard

Running Demo on iPhone Retina (4-inch) 4

Main.storyboard

Demo
2 targets, iOS SDK 7.0
Demo
DemoAppDelegate.h
DemoAppDelegate.m
PokemonsViewController.h
PokemonsViewController.m
PokemonCVCell.h
PokemonCVCell.m
models
icon
Main.storyboard
Images.xcassets
Supporting Files
DemoTests
Frameworks
Products

Pokemons View Cont...
Pokemons View Co...
Collection View
PokemonCV...
Label – Label
Image View
Constraints
Collection Vi...
First Responder
Exit

Label UI

Collection View Size

Cell Size 32 42

Header Size 0,0 0,0

Footer Size 0,0 0,0

Min Spacing 5 5

For Cells For Lines

Section Insets 0 0

Top Bottom

5 5

Left Right

Collection View Size

ScrollView Size

Scroll Indicators

Insets 0 0

Top Bottom

0 0

Left Right

Collection Reusable View - Defines the attributes and behavior of reusable views in a collection view, such as a...

Image View - Displays a single image, or an animation described by an array of images.

Text View - Displays multiple lines of editable text and sends an action message to a target object when...

Auto | © Santiago Pavón - UPM-DII

Demo.xcodeproj — Main.storyboard

Running Demo on iPhone Retina (4-inch) 4

Main.storyboard

Demo 2 targets, iOS SDK 7.0

Demo

- DemoAppDelegate.h
- DemoAppDelegate.m
- PokemonsViewController.h
- PokemonsViewController.m
- PokemonCVCell.h
- PokemonCVCell.m

models

icon

Main.storyboard

Images.xcassets

Supporting Files

DemoTests

Frameworks

Products

Pokemons View Cont... ▾

- Pokemons View Co... ▾
- Collection View ▾
- PokemonCV... ▾
- Label – Label
- Image View
- Constraints
- Collection Vi...
- First Responder
- Exit

Label

Text Plain

Label

Color Black Default

Font System 10.0

Alignment

Lines 1

Behavior Enabled Highlighted

Baseline Align Baselines

Line Breaks Truncate Tail

Autoshrink Minimum Font Size 7

Tighten Letter Spacing

Highlighted Black Default

Shadow

Shadow Offset 0 -1

Horizontal Vertical

Collection Reusable View – Defines the attributes and behavior of reusable views in a collection view, such as a...

Image View – Displays a single image, or an animation described by an array of images.

Text View – Displays multiple lines of editable text and sends an action message to a target object when...

Auto

© Santiago Pavón - UPM-DIT

The screenshot shows the Xcode interface with the storyboard editor open. A red box highlights the 'Attributes Inspector' on the right side of the screen, which is displaying settings for a selected 'Label' object. The label's text is set to 'Plain' and its font is 'System 10.0'. The 'Tighten Letter Spacing' checkbox is checked. Below the inspector, there are three cards providing information about 'Collection Reusable View', 'Image View', and 'Text View'.

Demo.xcodeproj — Main.storyboard

Finished running Demo on iPhone Retina (3.5-inch) 4

Main.storyboard

Demo

2 targets, iOS SDK 7.0

Demo

DemoAppDelegate.h

DemoAppDelegate.m

PokemonsViewController.h

PokemonsViewController.m

PokemonCVCell.h

PokemonCVCell.m

models

icon

Main.storyboard

Images.xcassets

Supporting Files

DemoTests

Frameworks

Products

Pokemons View Cont...

Pokemons View Co...

Collection View

Collection Re...

PokemonCV...

Label – Label

Image View

Constraints

Collection Vi...

First Responder

Exit

Collection View

Items 1

Layout Flow

Scroll Direction Vertical

Accessories Section Header Section Footer

Style Default

Shows Horizontal Indicator Shows Vertical Indicator

Scrolling Scrolling Enabled Paging Enabled Direction Lock Enabled

Bounce Bounces Bounce Horizontally Bounce Vertically

Zoom 0 Min 0 Max

Añadir cabeceras a las secciones.

Collection Reusable View – Defines the attributes and behavior of reusable views in a collection view, such as a...

Image View – Displays a single image, or an animation described by an array of images.

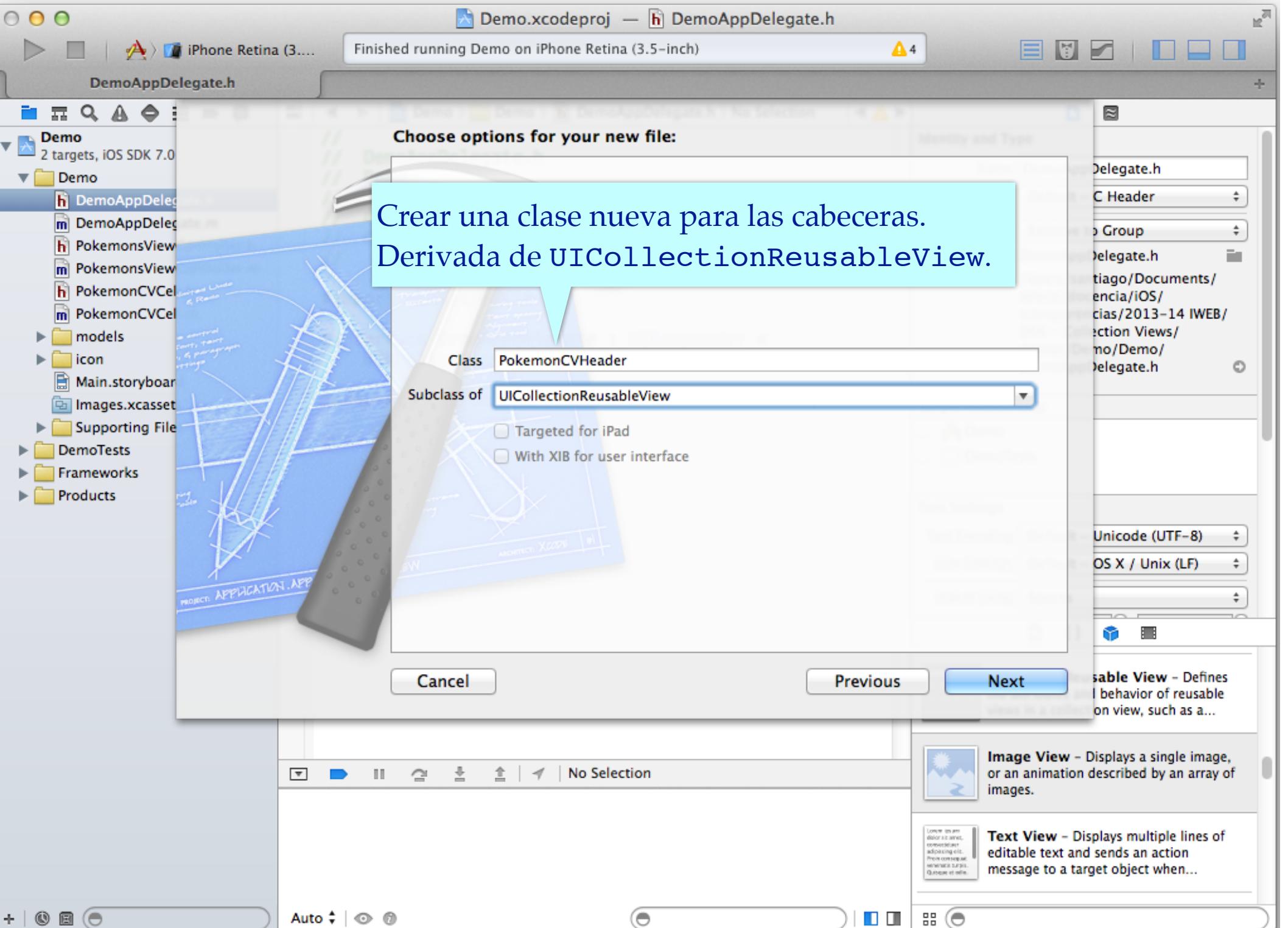
Text View – Displays multiple lines of editable text and sends an action message to a target object when...

No Selection

Auto

© Santiago Pavón - UPM-DIT

The screenshot shows the Xcode interface with the project 'Demo' open. In the storyboard, a 'Collection View' is selected. A red circle highlights the top section of the collection view, which contains a 'Section Header' (labeled 'Label') and a 'Section Footer' (labeled 'UII'). A callout bubble with the text 'Añadir cabeceras a las secciones.' points to this highlighted area. The Attributes Inspector on the right shows various properties for the collection view, including 'Items' set to 1, 'Layout' set to 'Flow', 'Scroll Direction' set to 'Vertical', and 'Accessories' with 'Section Header' checked. Other options like 'Shows Horizontal Indicator' and 'Shows Vertical Indicator' are also checked. The bottom of the screen shows the Xcode toolbar with icons for file operations, search, and selection.



Cambiar la clase de la cabecera
a PokemonCVHeader.

The screenshot shows the Xcode interface with the storyboard editor open. A blue callout points from the text above to the 'Identity' section of the right-hand Utilities pane. In the storyboard, a collection view header is selected, and its class is set to 'PokemonCVHeader' in the Identity inspector. The file 'PokemonCVHeader.h' is circled in red in the Project Navigator.

Project Navigator:

- Demo (2 targets, iOS SDK 7.0)
 - Demo
 - DemoAppDelegate.h
 - DemoAppDelegate.m
 - PokemonsViewController.h
 - PokemonsViewController.m
 - PokemonCVCell.h
 - PokemonCVCell.m
 - PokemonCVHeader.h** (circled in red)
 - PokemonCVHeader.m**
 - models
 - icon

Main.storyboard

File Navigator:

 - Images.xcassets
 - Supporting Files
 - DemoTests
 - Frameworks
 - Products

Storyboard Editor:

Pokemons View Controller... (selected)

 - Pokemons View Controller
 - Collection View
 - PokemonCV Header
 - PokemonCV Cell -...
 - Collection View Flo...
 - First Responder
 - Exit

Identity Inspector:

Custom Class: PokemonCVHeader

Identity

User Defined Runtime Attributes

Document

 - Label: Xcode Specific Label
 - Object ID: uyM-GA-6qL
 - Lock: Inherited - (Nothing)
 - Notes: No Font

Callout Content:

 - Collection Reusable View** - Defines the attributes and behavior of reusable views in a collection view, such as a...
 - Image View** - Displays a single image, or an animation described by an array of images.
 - Text View** - Displays multiple lines of editable text and sends an action message to a target object when...

Demo.xcodeproj — Main.storyboard

Finished running Demo on iPhone Retina (3.5-inch) ⚠ 2

Main.storyboard

Pokemons View Controller...

- Pokemons View Controller
 - Collection View
 - PokemonCV Header
 - Label - Name
 - Label - # Races
 - Constraints
 - Vertical Spac...
 - Vertical Spac...
 - Horizontal S...
 - Vertical Spac...
 - Vertical Spac...
 - Horizontal S...
 - Horizontal S...
 - PokemonCV Cell ...
 - Collection View Flo...
- First Responder
- Exit

Name # Races

Label UII

Diseñar la cabecera.

Label

- Text Plain
- Name
- Color Black Color
- Font System Bold 20.0
- Alignment
- Lines 1
- Behavior Enabled
- Highlighted
- Baseline Align Baselines
- Line Breaks Truncate Tail
- Autoshrink Fixed Font Size
- Tighten Letter Spacing
- Highlighted Default
- Shadow Default
- Shadow Offset 0 -1
- Horizontal Vertical

Interface Builder.

Label Label - A variably sized amount of static text.

Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Demo.xcodeproj — Main.storyboard

Finished running Demo on iPhone Retina (3.5-inch) 2

Main.storyboard

Demo > M... > M... > Po... > Po... > Collection View > PokemonCV Header

Pokemons View Cont...
Pokemons View Co...
Collection View
PokemonCV...
Label - Na...
Label - #...
Constraints
PokemonCV...
Collection Vi...
First Responder
Exit

Name # Races

Label UII

Collection Reusable View
Identifier TypesHeader

Mode Center
Tag 0

Interaction User Interaction Enabled
Multiple Touch

Alpha 1

Background Default
Tint Default

Drawing Opaque
Hidden
Clears Graphics Context
Clip Subviews
Autosize Subviews

Stretching 0 0
X Y
Width Height

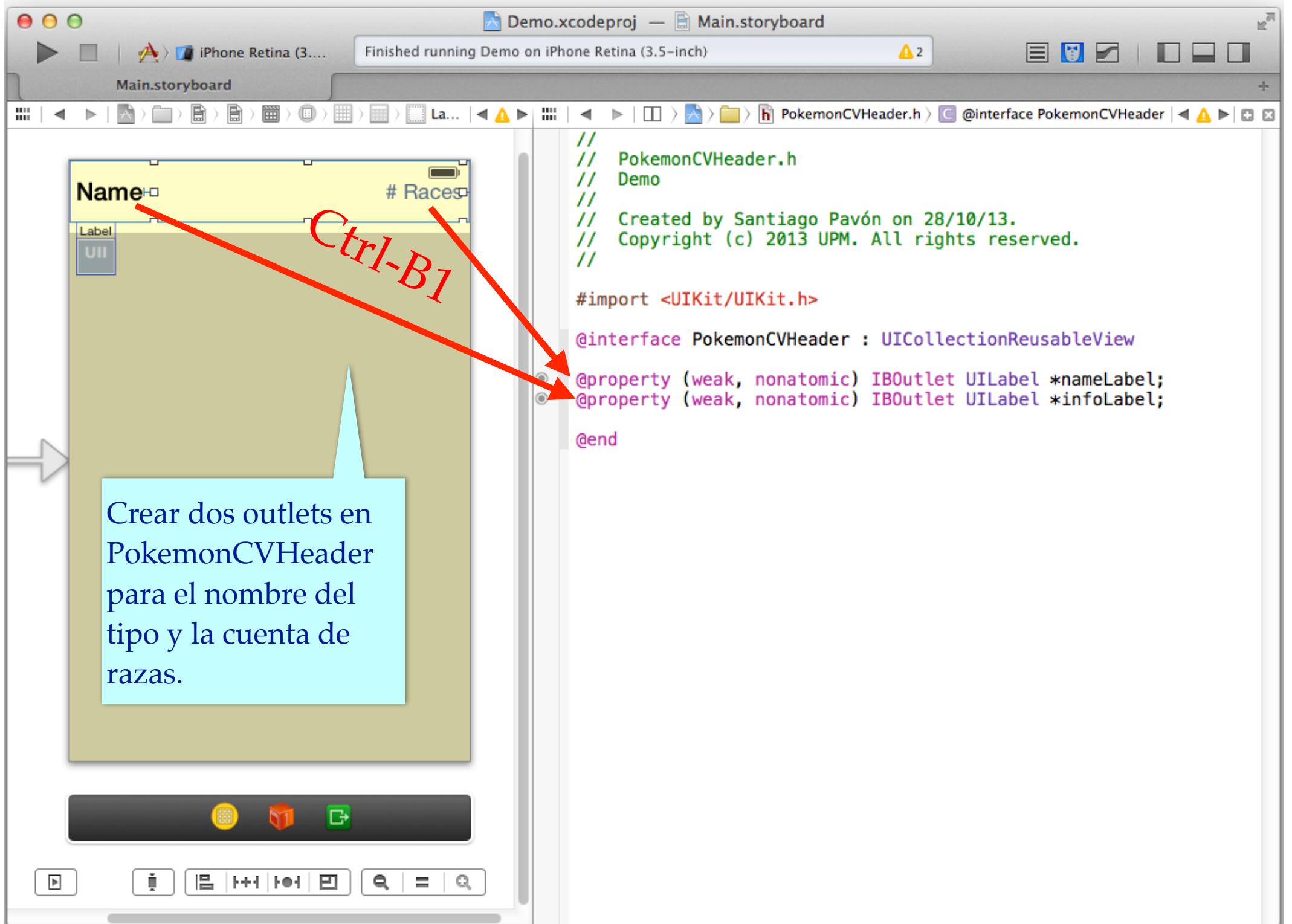
Interface Builder.

Label Label - A variably sized amount of static text.

Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Poner un identificador de reutilización a la view de cabecera.



Demo.xcodeproj — PokemonsViewController.m

Finished running Demo on iPhone Retina (3.5-inch) No Issues

PokemonsViewController.m

Demo > Demo > PokemonsViewController.m > No Selection

```
#pragma mark - Collection View Data Source

- (NSInteger) numberOfSectionsInCollectionView:(UICollectionView *)collectionView
{ [...] }

- (NSInteger) collectionView:(UICollectionView *)collectionView
    numberOfItemsInSection:(NSInteger)section
{ [...] }

- (UICollectionViewCell *)collectionView:(UICollectionView *)collectionView
    cellForItemAtIndexPath:(NSIndexPath *)indexPath
{ [...] }

-(UICollectionViewReusableView *)collectionView:(UICollectionView *)collectionView
    viewForSupplementaryElementOfKind:(NSString *)kind
    atIndexPath:(NSIndexPath *)indexPath
{
    if ([kind isEqualToString:UICollectionViewElementKindSectionHeader]) {

        PokemonCVHeader *header = [collectionView dequeueReusableCellWithReuseIdentifier:@"TypesHeader"
                                forIndexPath:indexPath];
        Type * type = self.pokedexModel.types[indexPath.section];

        header.nameLabel.text = [type.name capitalizedString];
        header.infoLabel.text = [NSString stringWithFormat:@"%d razas", [type.races count]];

        return header;
    }

    return nil;
}

@end
```

Método del Data Source

Añadir arriba:
#import "PokemonCVHeader.h"



Demo.xcodeproj — Main.storyboard

iPhone Retina (3....) Finished running Demo on iPhone Retina (3.5-inch) No Issues

Demo

Demo

DemoAppDelegate.h

DemoAppDelegate.m

PokemonsViewController.h

PokemonsViewController.m

PokemonCVCell.h

PokemonCVCell.m

PokemonCVHeader.h

PokemonCVHeader.m

WebViewController.h

WebViewController.m

models

icon

Main.storyboard

Images.xcassets

Supporting Files

DemoTests

Frameworks

Products

Navigation Controller

Usar un Navigation Controller y otra escena con un UIWebView para mostrar información descargada de internet.
Similar al ejemplo 065-Pokedesk.

```
graph LR; NC[Navigation Controller] --> PV[Pokemons View Controller]; PV --> WC[Web View Controller]
```

The storyboard diagram illustrates a navigation flow. It begins with a "Navigation Controller" (represented by a blue-bordered box), which pushes to a "Pokemons View Controller" (represented by a yellow-bordered box). This controller contains a table view with columns "Name" and "# Races". Finally, it pushes to a "Web View Controller" (represented by a light blue-bordered box).

Demo.xcodeproj — PokemonsViewController.m

iPhone Retina (3....) Finished running Demo on iPhone Retina (3.5-inch) No Issues

Demo 2 targets, iOS SDK 7.0

Demo

- DemoAppDelegate.h
- DemoAppDelegate.m
- PokemonsViewController.h
- PokemonsViewController.m**
- PokemonCVCell.h
- PokemonCVCell.m
- PokemonCVHeader.h
- PokemonCVHeader.m
- WebViewController.h
- WebViewController.m
- models
- icon
- Main.storyboard
- Images.xcassets
- Supporting Files
- DemoTests
- Frameworks
- Products

```
#pragma mark - Navigation

// In a story board-based application, you will often want to do a little preparation
// before navigation
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    if ([segue.identifier isEqualToString:@"Show Web Info"]) {
        NSIndexPath * ip = [self.collectionView indexPathForCell:sender];
        WebViewController *wvc = segue.destinationViewController;
        Type * type = self.pokedeskModel.types[ip.section];
        wvc.race = type.races[ip.item];
    }
}

@end
```

