



# Desenvolvimento para iPhone

Usando iOS SDK 8 e  
Objective-C

---

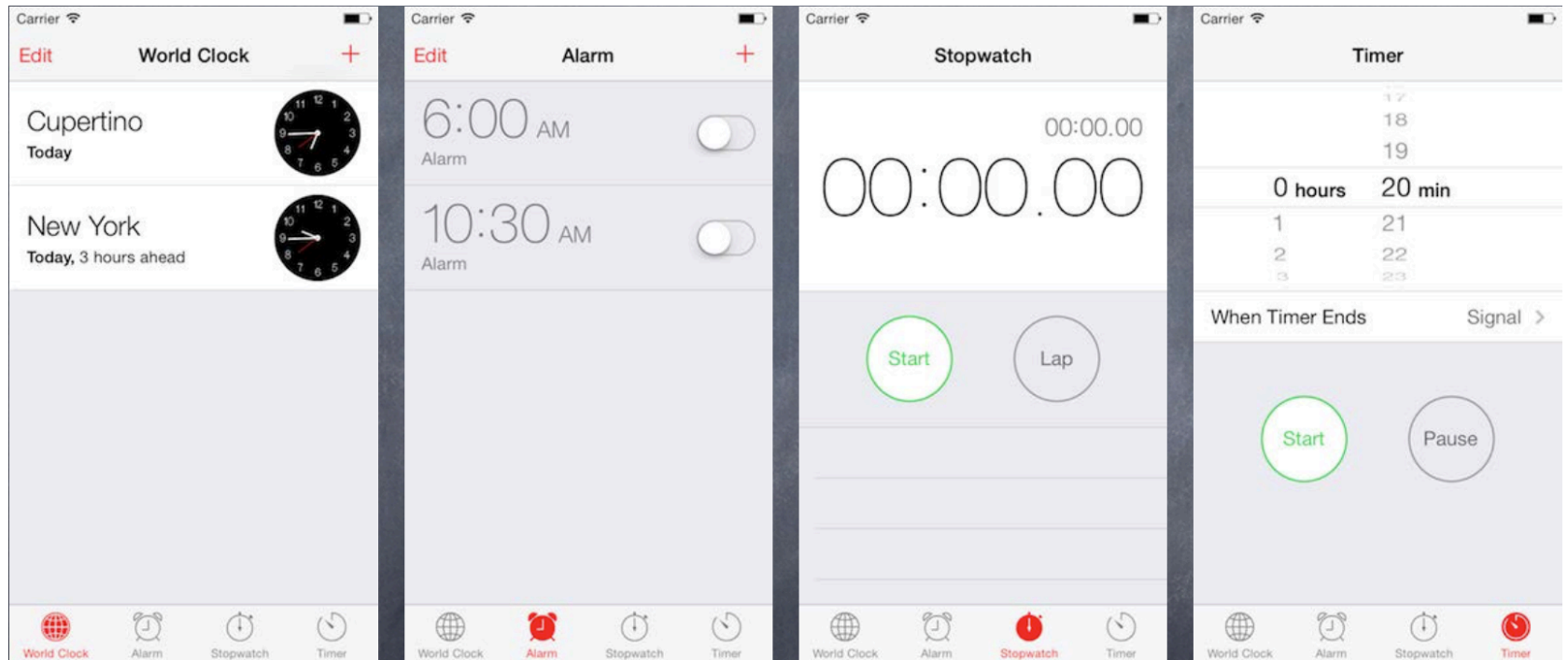
# Agenda

- UITabBarController;
- UICollectionViewController;
- Reconhecimento de Gestos;
- Introdução ao Autolayout;
  - Conceitos básicos de coordenadas e posicionamento de elementos;

# Objetivos do Dia

- Compreender o funcionamento do UITabBarController;
- Entender o UICollectionViewController;
- Conseguir implementar reconhecimento de gestos pré-definidos;
- Começar o estudo do Autolayout.

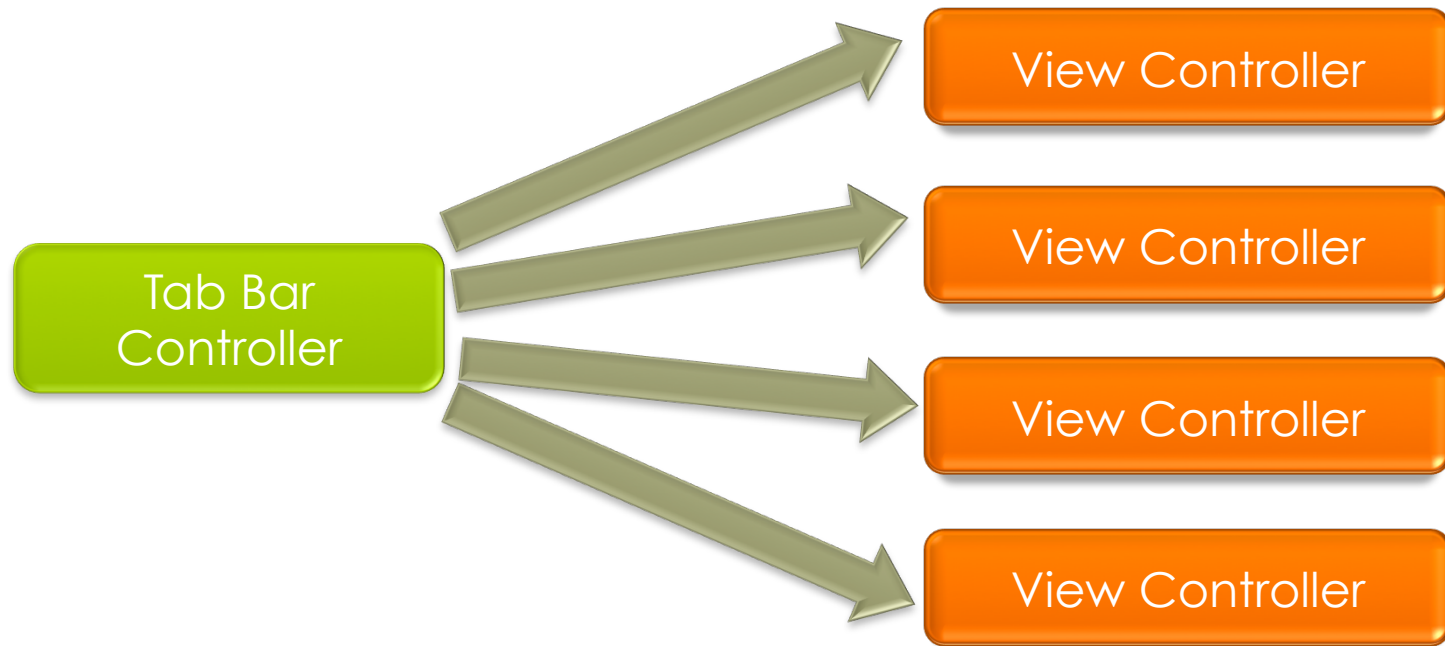
# UITabBarController



# UITabBarController

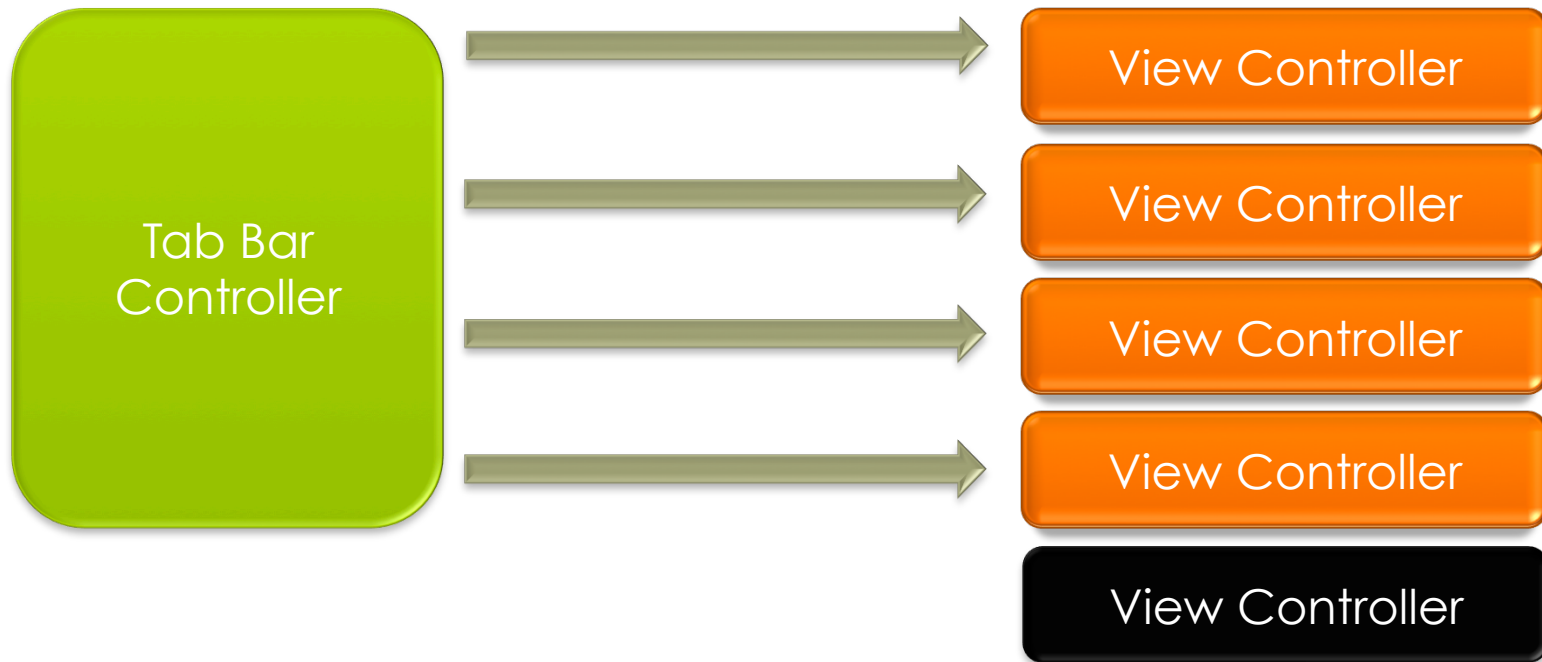
- Um dos principais componentes do iOS;
- Controla abas, onde cada aba é um UIViewController;
- Exibe o número de abas que o dispositivo é capaz de exibir sem comprometer o design;
- Cuidado!
  - O UITabBarController não pode estar na hierarquia de ViewControllers do UINavigationController!
  - O oposto é o correto, portanto.

# UITabBarController

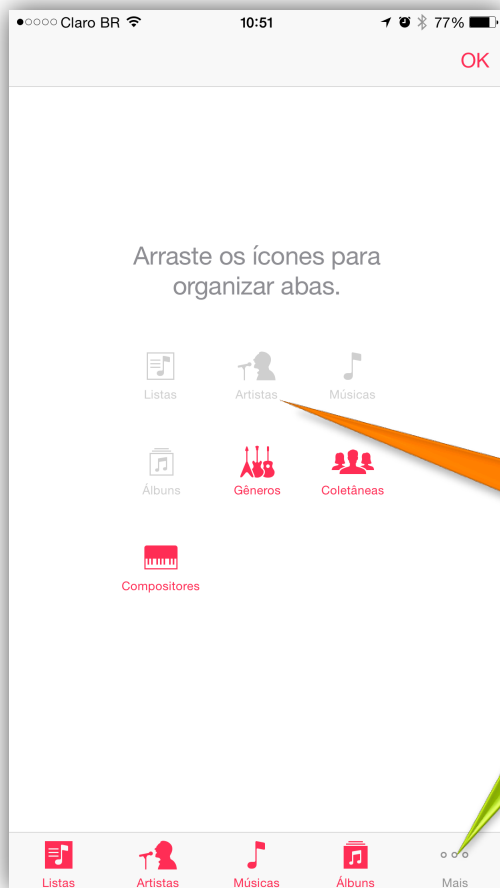


# UITabBarController

- E se eu precisar de mais abas?



# UITabBarController



Quando há mais do que 4 ViewControllers associados, o botão “Mais” aparece automaticamente.

Esta interface, que permite ao usuário reordenar as abas, já vem pronta e aparece automaticamente quando há mais de 4 abas!



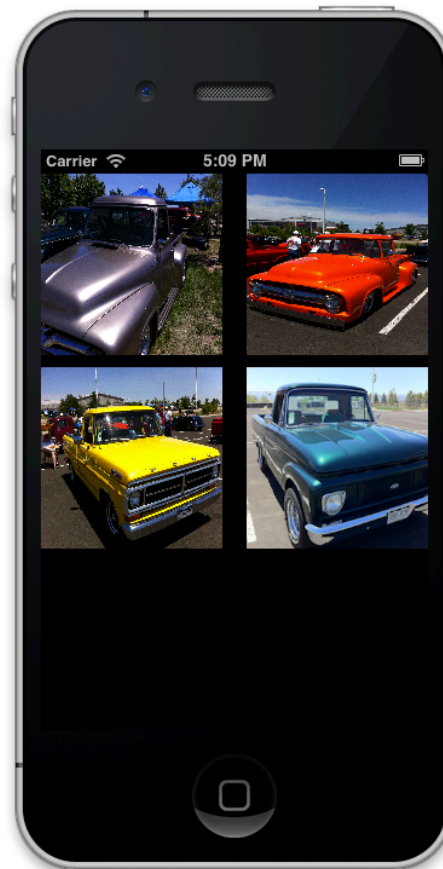
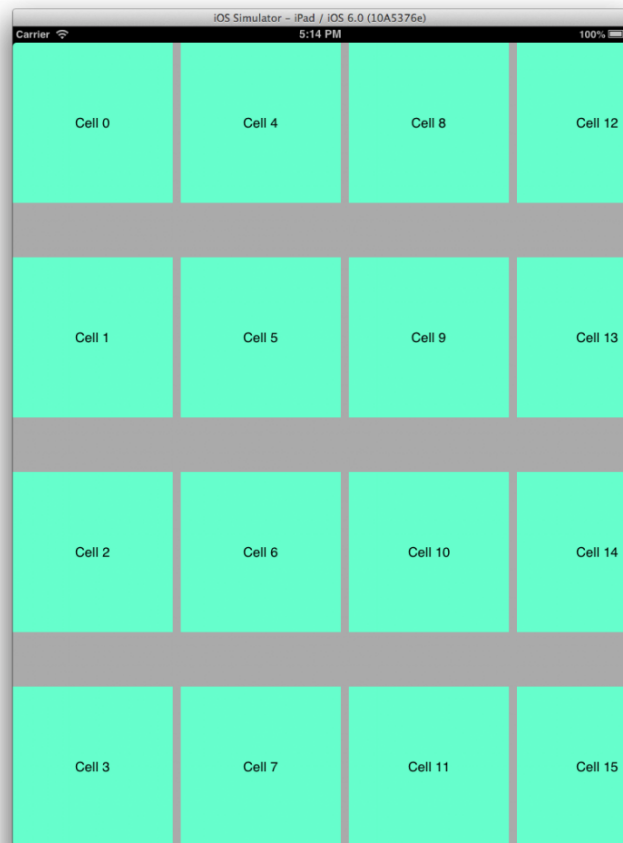
# UICollectionViewController

- Irmão do UITableViewController;
- Serve para exibir um *array* de views em um formato que não seja de lista;
- Um dos usos mais comuns é a construção de galerias de fotos;
- A organização mais usada é o formato de *grid*.

# UICollectionViewController

- UICollectionViewCell;
  - E subclasses customizadas;
- UICollectionViewDataSource;
- UICollectionViewDelegate;
- UICollectionViewLayout;
  - UICollectionViewFlowLayout

# UICollectionViewController



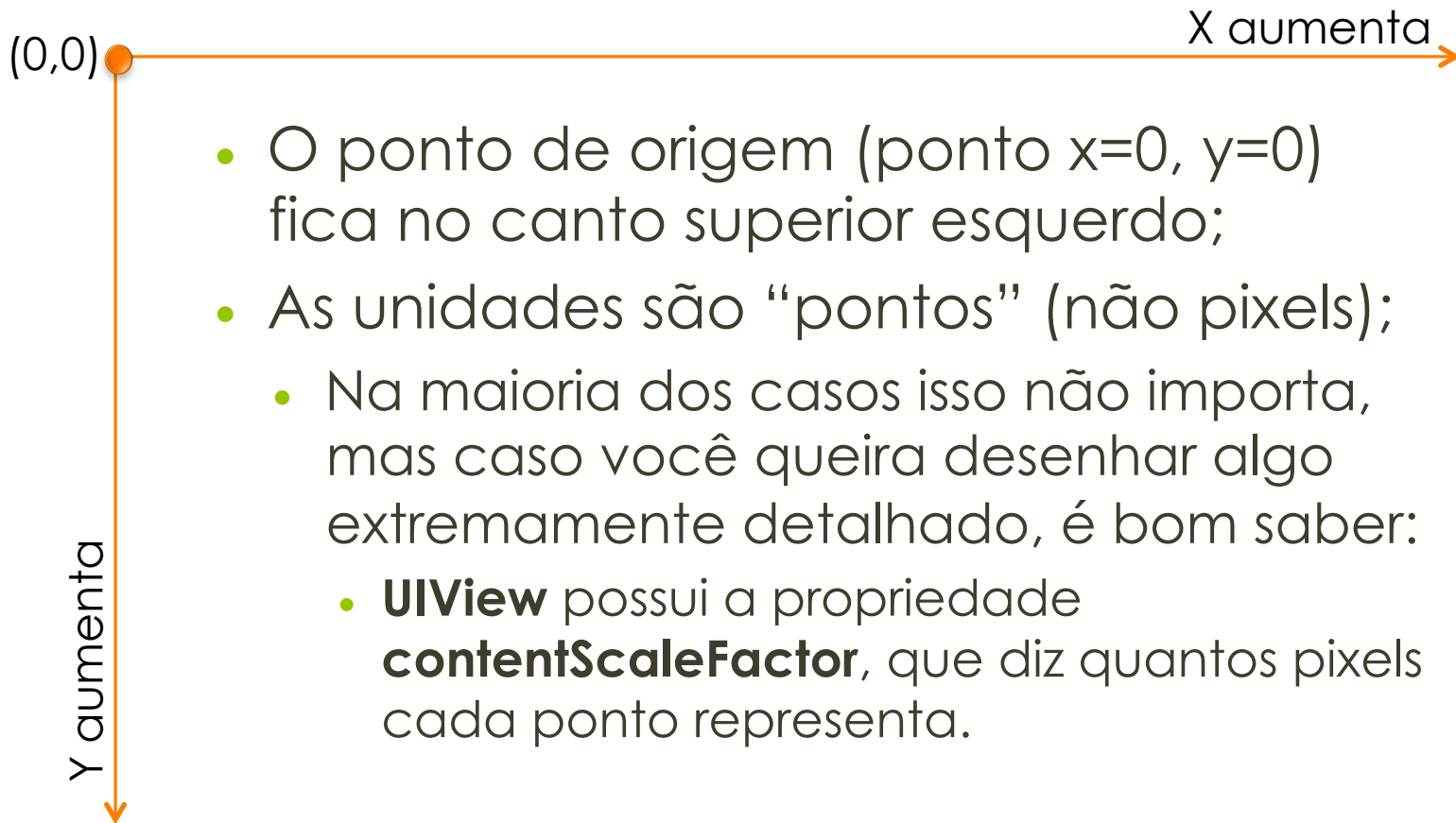
# Reconhecimento de Gestos

- Tap;
  - Toque (com um ou mais dedos)
- Swipe;
- Pan;
  - Arrastar
- Pinch;
- Rotation;
- <http://www.appcoda.com/ios-gesture-recognizers/>

# Introdução ao Autolayout

- Sistema de coordenadas;
- Sistema de unidades;
- Structs envolvidas;
  - Parte do framework CoreGraphics – inteiramente em linguagem C;

# Introdução ao Autolayout



# Introdução ao Autolayout

- Mais propriedades interessantes:
- **@property CGRect** bounds;
  - O espaço interno e tamanho da sua View. É usada para a implementação interna da View.
- **@property CGPoint** center;
  - O centro da sua View no sistema de coordenadas da SuperView.
- **@property CGRect** frame;
  - Um retângulo na SuperView que contém inteiramente a sua view (bounds.size)

# Introdução ao Autolayout

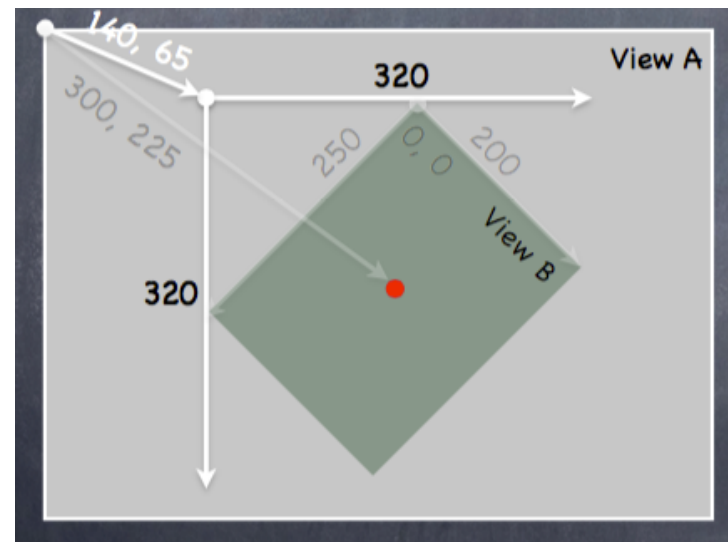


- Views podem passar por transformações (escala, rotação e translação), portanto cuidado:
  - bounds** de B = ((0,0), (250,250))
  - frame** de B = ((140,65), (320,320))
  - center** de B = (300,225)



# Introdução ao Autolayout

- Views raramente são rotacionadas, mas tome cuidado para não fazer confusão com **frame** e **center**!



# Introdução ao Autolayout

- Por padrão, quando o tamanho de uma view (**bounds**) muda, não ocorre redesenho;
  - Em vez disso, a view é esticada, encolhida ou ainda reposicionada.
- Eventualmente pode não ser esse o comportamento que você deseja...
  - **@property UIViewContentMode** contentMode;

# Introdução ao Autolayout

- **UIViewContentMode**  
{Left,Right,Top,Right,BottomLeft,BottomRight,TopLeft,TopRight}
  - Move a view para a respectiva localização
- **UIViewContentModeScale**  
{ToFill,AspectFill,AspectFit}
  - Escala a view (estica ou encolhe)
- **UIViewContentModeRedraw**
  - Redesenha a view.
- Qual o padrão? **UIViewContentModeScaleToFill**

# Introdução ao Autolayout

- O que vem por aí?
- Na aula do dia 18/06, vamos aprender a:
  - Lidar com frames usando regras em vez de números com uma API muito poderosa (NSLayoutConstraint)!
  - Mas, desde o Xcode 5, podemos fazer a maior parte do trabalho graficamente no InterfaceBuilder.

# Hora de Brincar!

- UITabBarController;
- UICollectionViewController;
- Gestos;
- Desafio!

# Desafio

- Navegar de uma tela para outra através do reconhecimento de um gesto.