

Desenvolvimento para iPhone

Usando Objective-C e iOS
SDK 8

Agenda

- Autolayout (parte 2);
- UIScrollView;
- Multithreading;
- Prática.

Objetivos do Dia

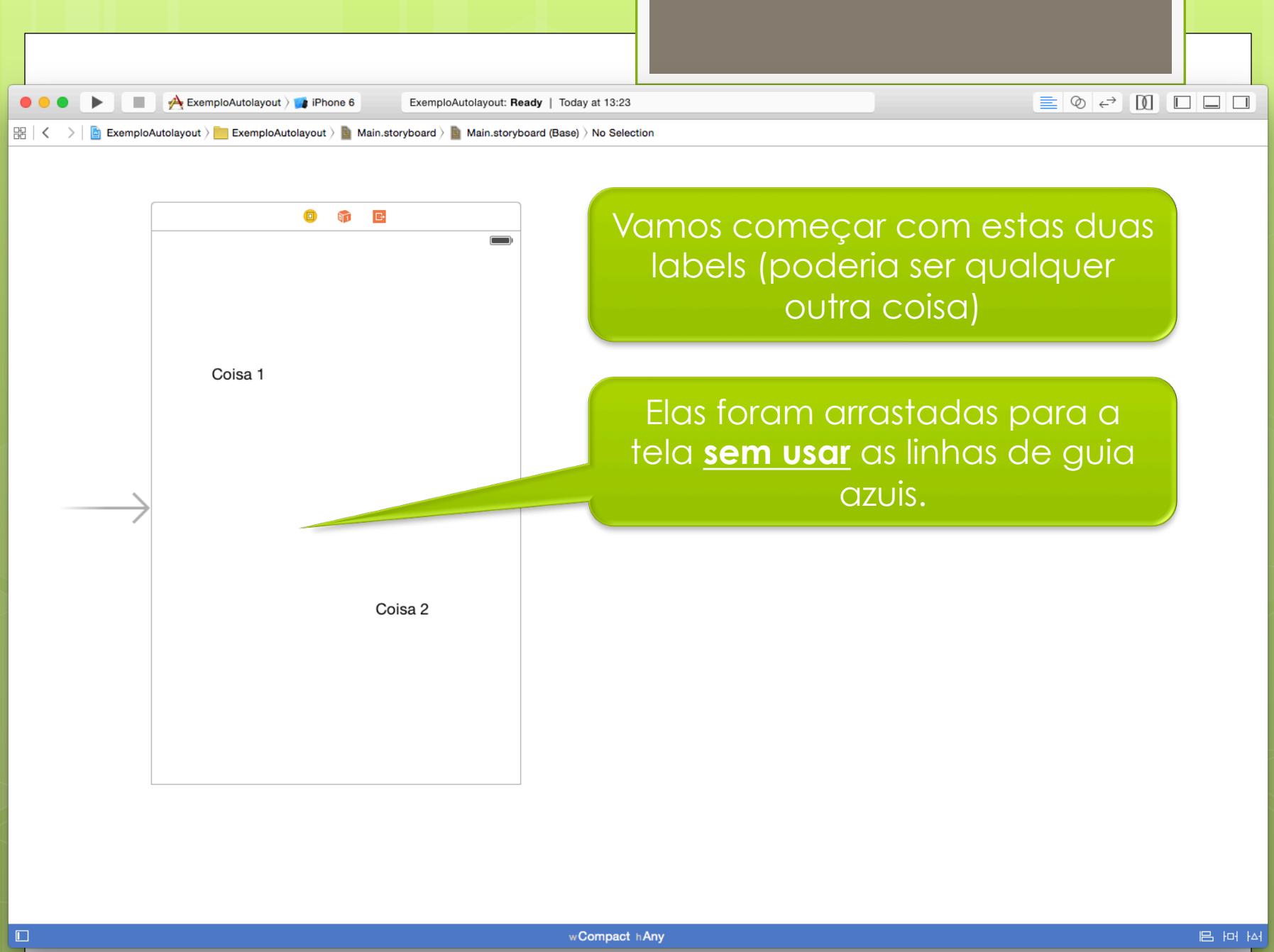
- Dominar o autolayout;
- Aprender a usar o UIScrollView;
- Expandir os conhecimentos sobre multithreading no iOS.

Autolayout

- “Configurar” tamanho e posição das views usando regras, em vez de números;
- Essas regras, quando bem definidas, fazem as coisas aparecerem nos lugares certos, não importa o tamanho que está disponível para o `ViewController`;
- As regras são denominadas **constraints**;
- Existe uma API muito poderosa para lidar com elas (**`NSLayoutConstraint`**), mas...

Autolayout

- Quase sempre, as regras são configuradas graficamente através do Xcode!



ExemploAutolayout > iPhone 6 ExemploAutolayout: Ready | Today at 13:23

ExemploAutolayout > ExemploAutolayout > Main.storyboard > Main.storyboard (Base) > No Selection

Coisa 1

Coisa 2

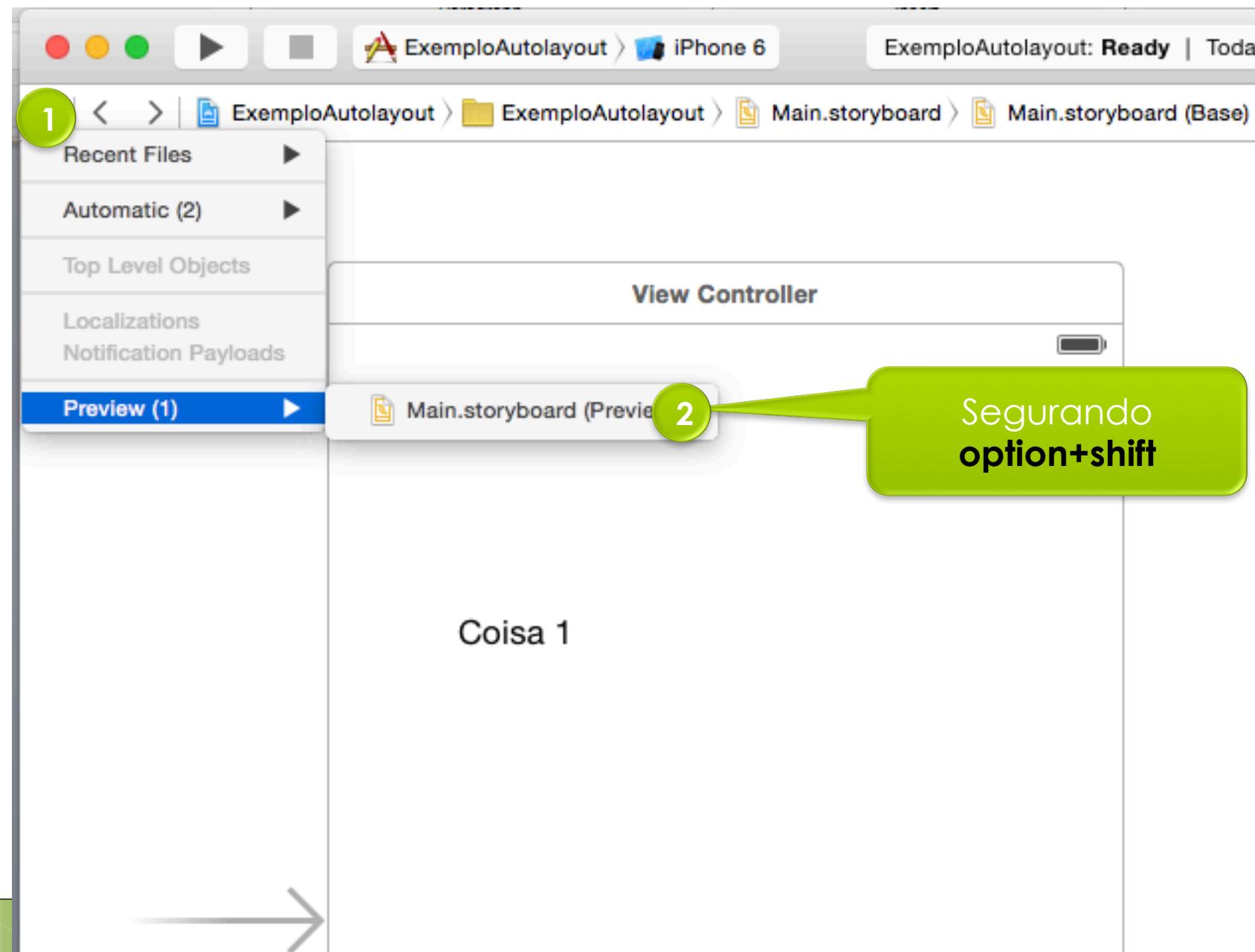
Seria muito bom que tivéssemos um jeito de ter uma visualização prévia do que vai acontecer em diferentes tamanhos de tela e em diferentes rotações...

O Xcode também acha! Por isso existe uma funcionalidade chamada Preview!

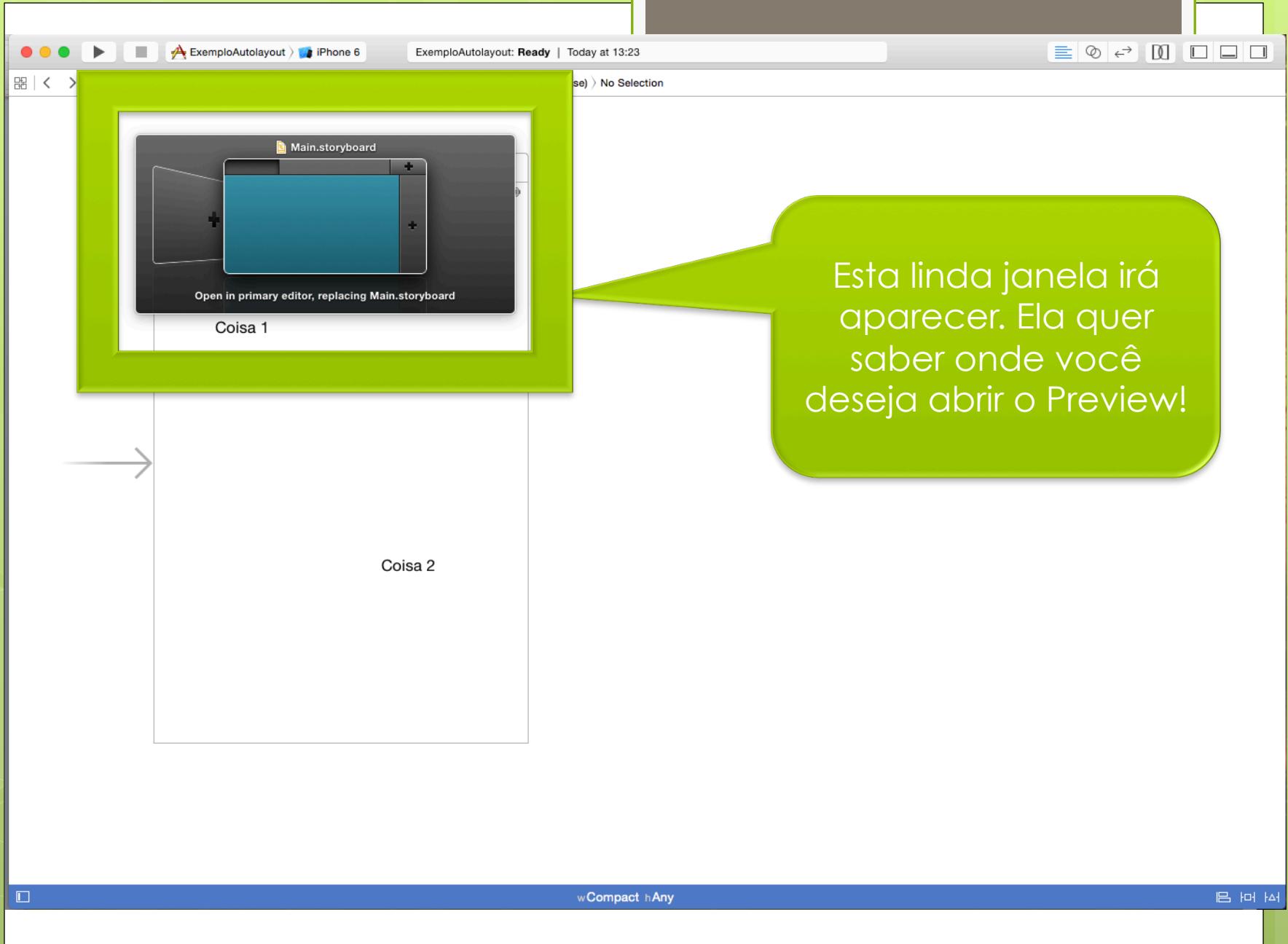
Você pode abrir o Preview onde quiser: editor assistente, mesma janela, outra janela



DO IT!
DO IT NOW!

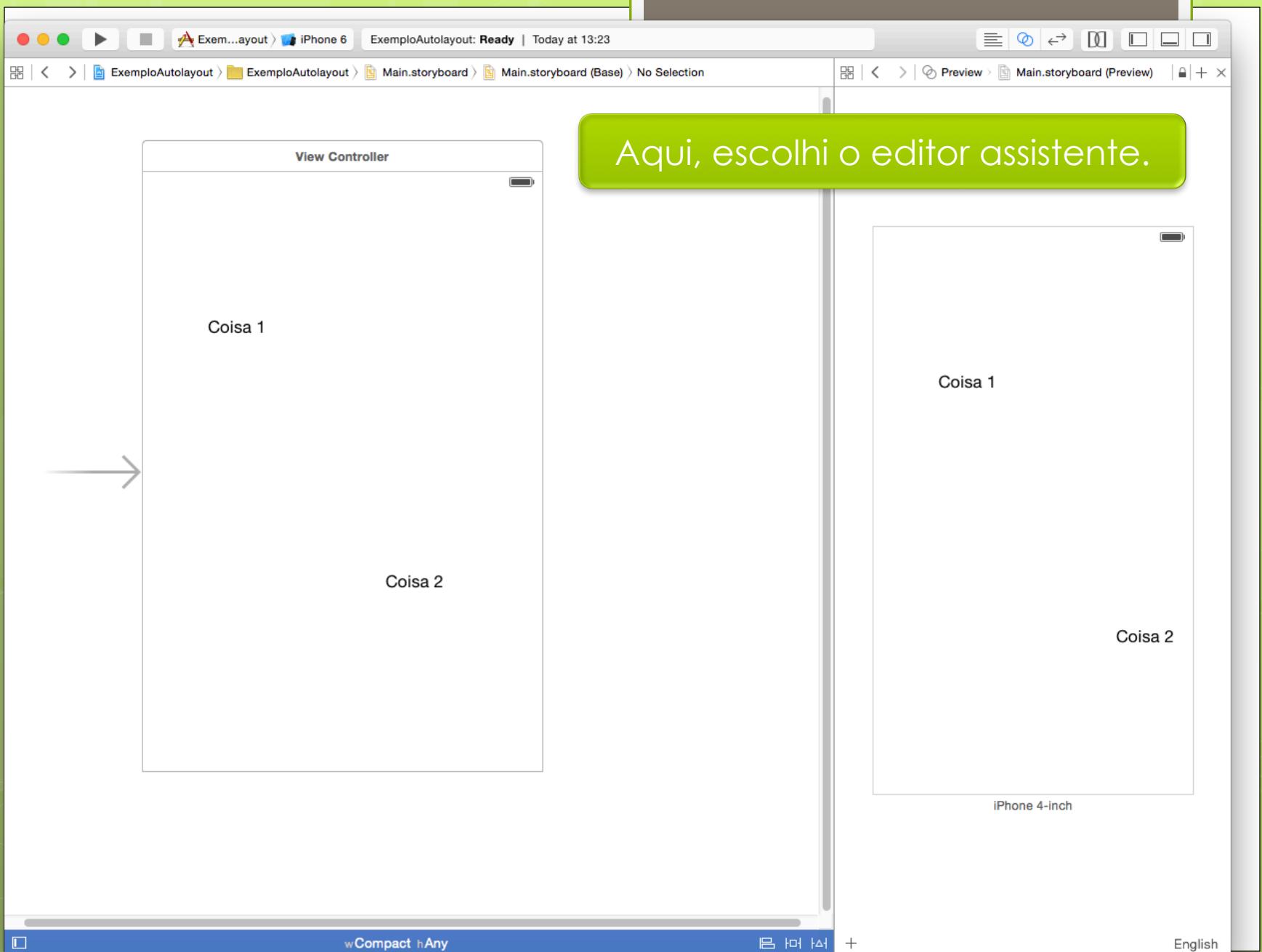


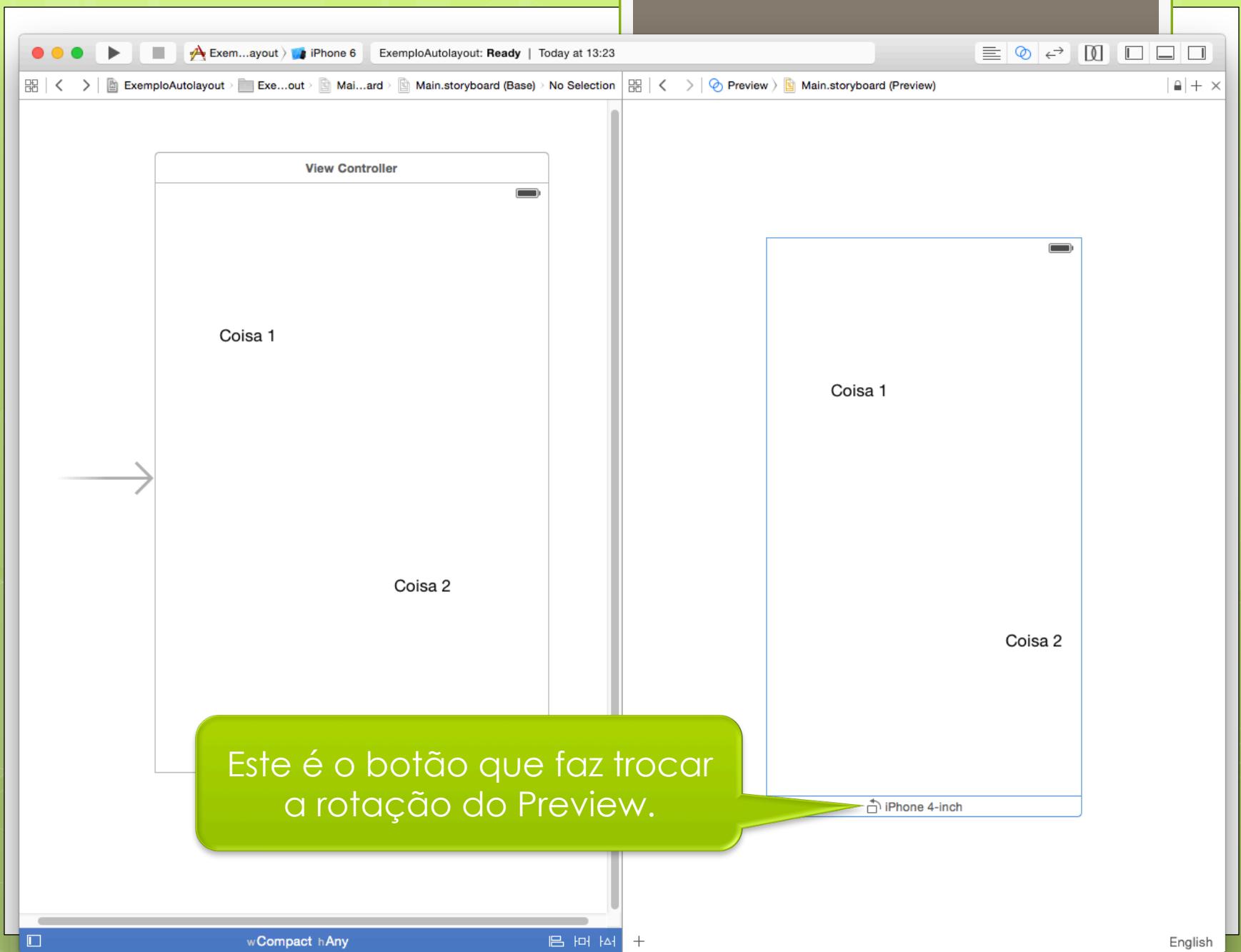
Segurando
option+shift

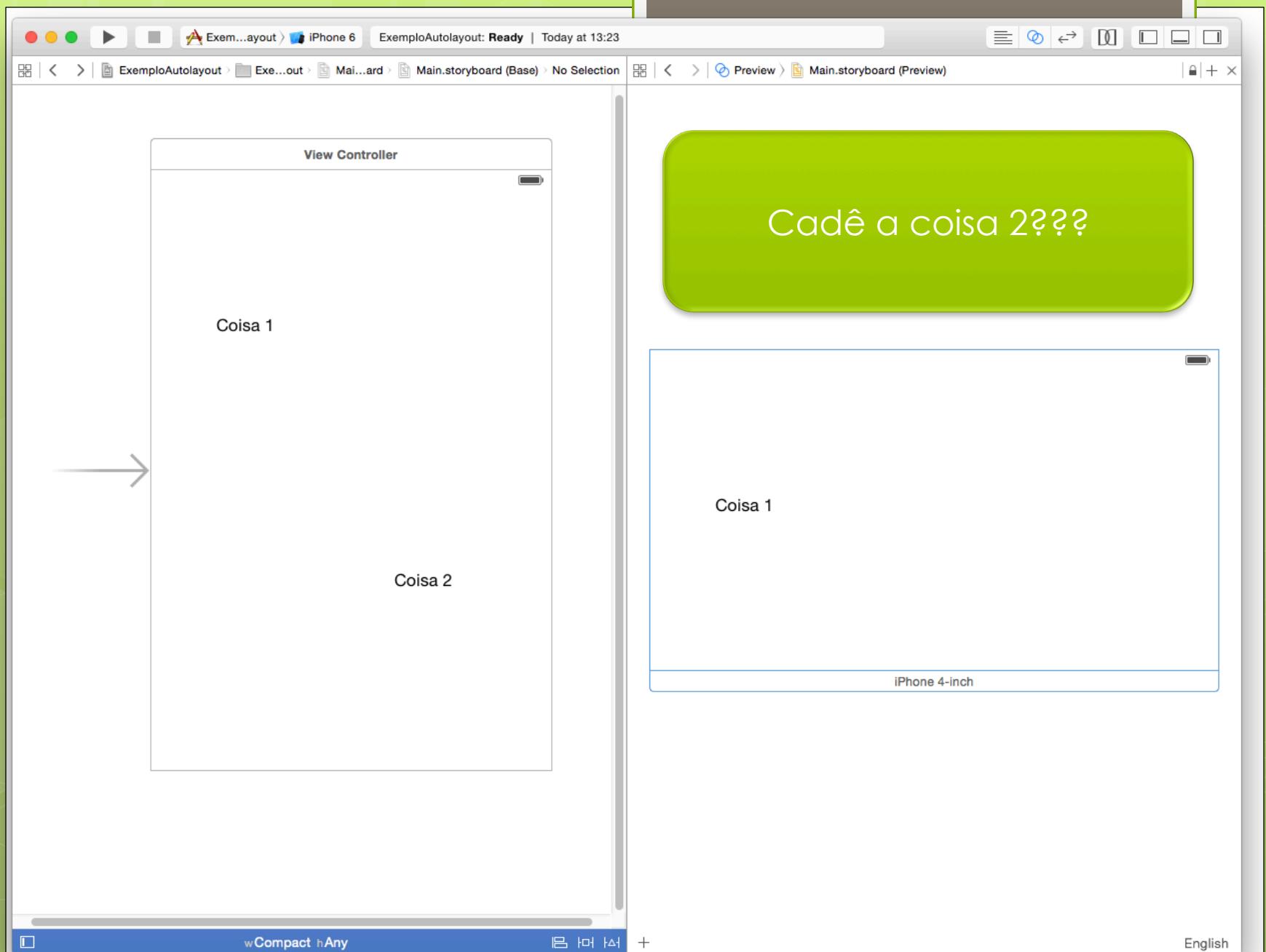


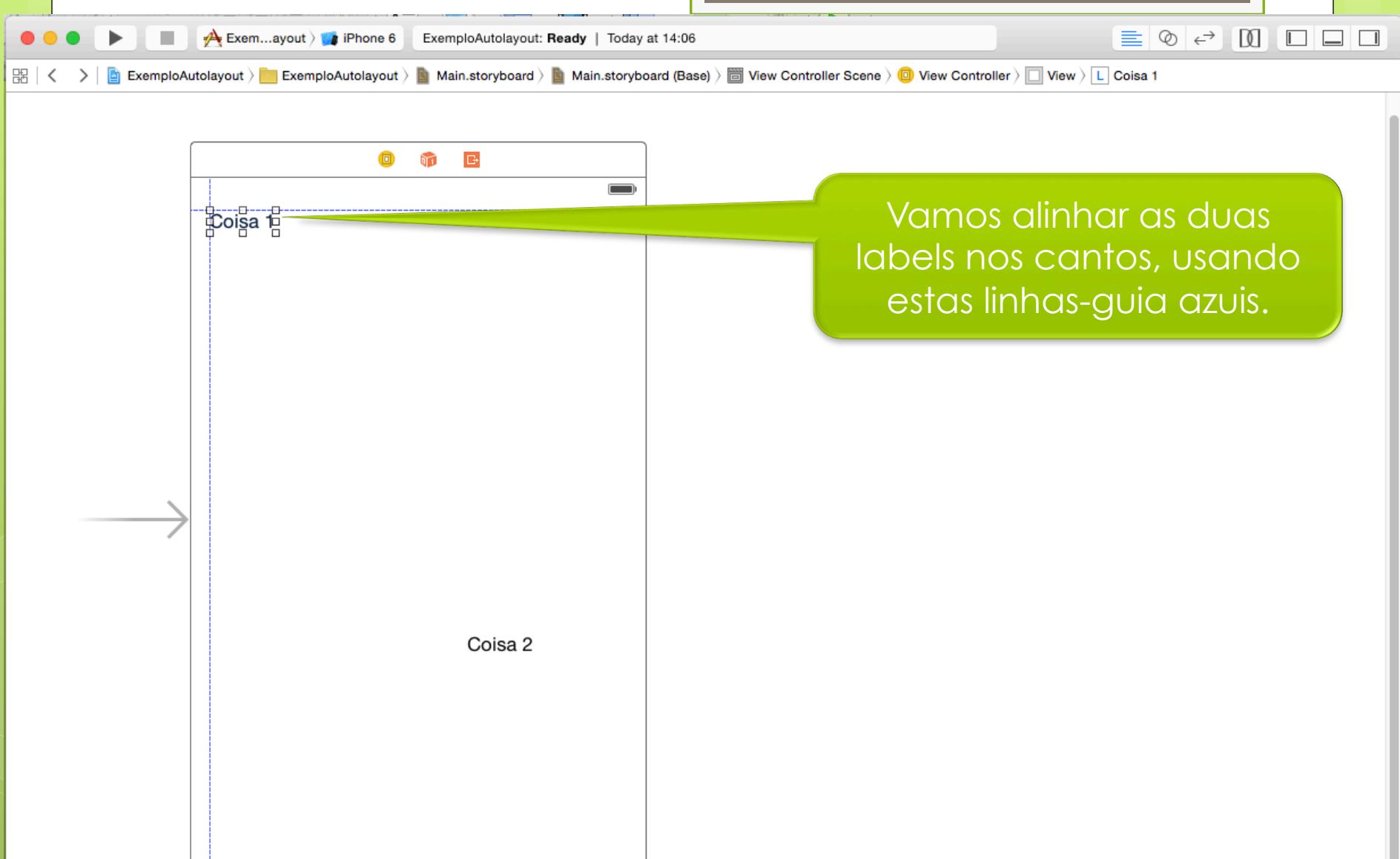
Esta linda janela irá aparecer. Ela quer saber onde você deseja abrir o Preview!

Aqui, escolhi o editor assistente.

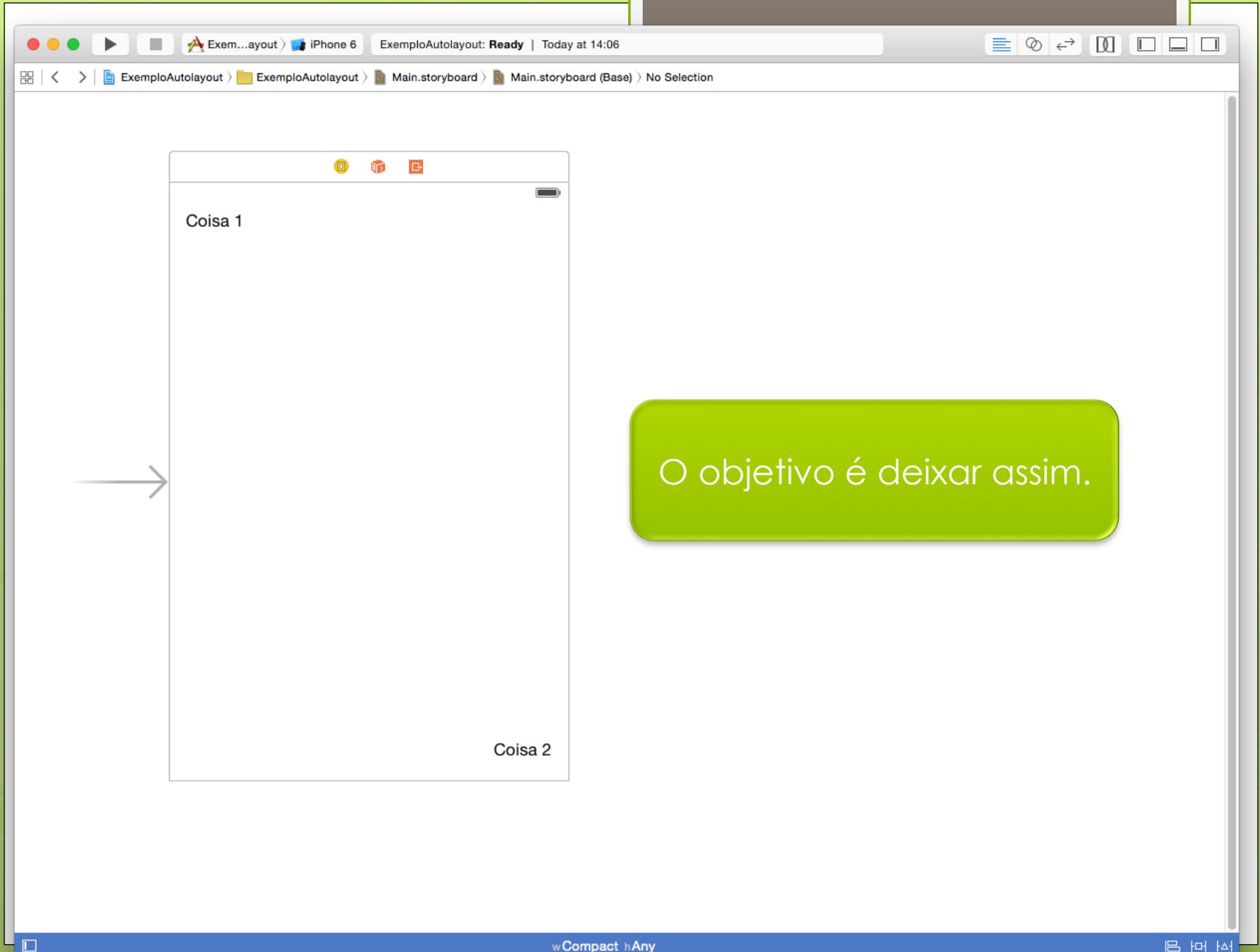


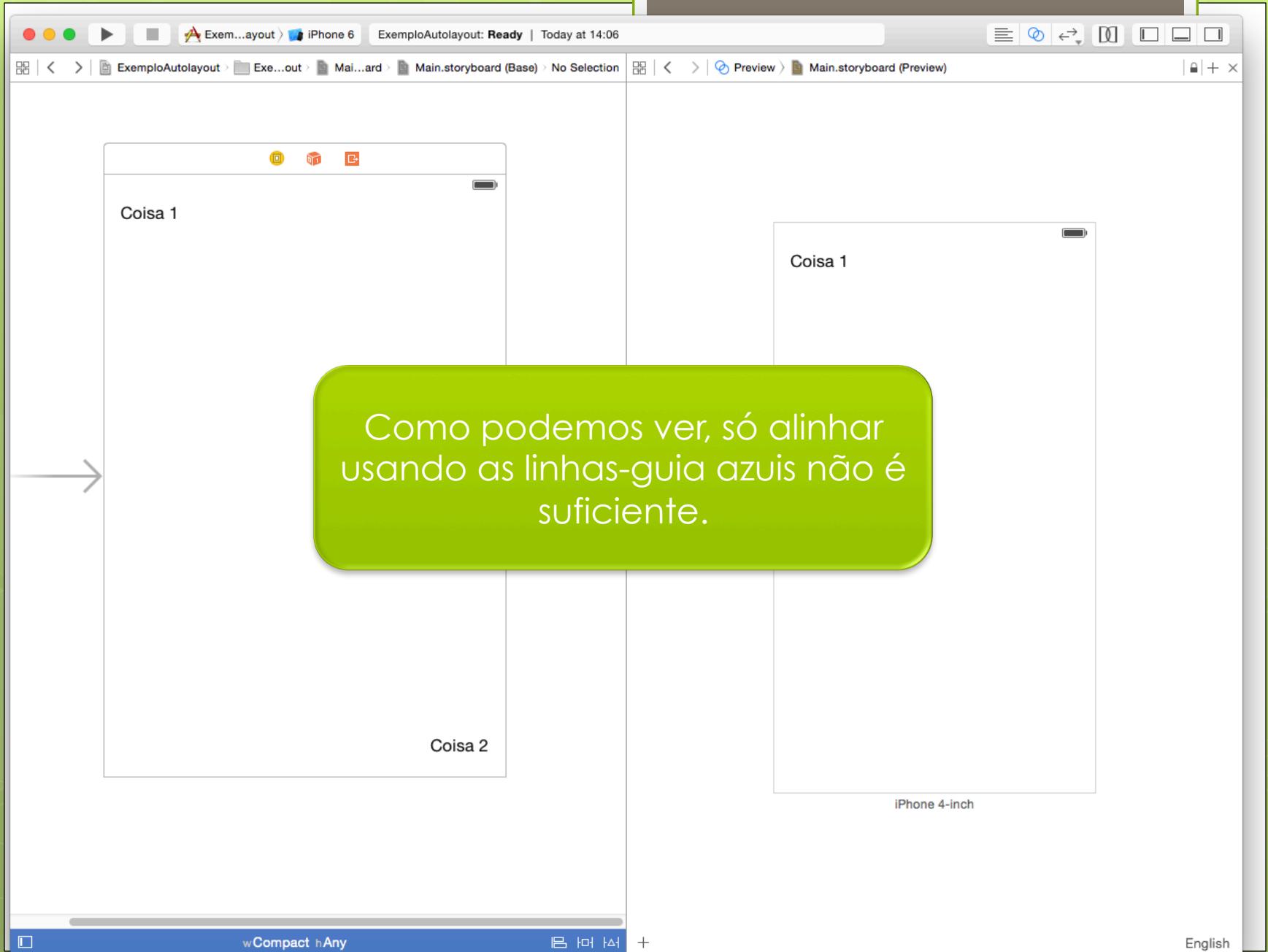






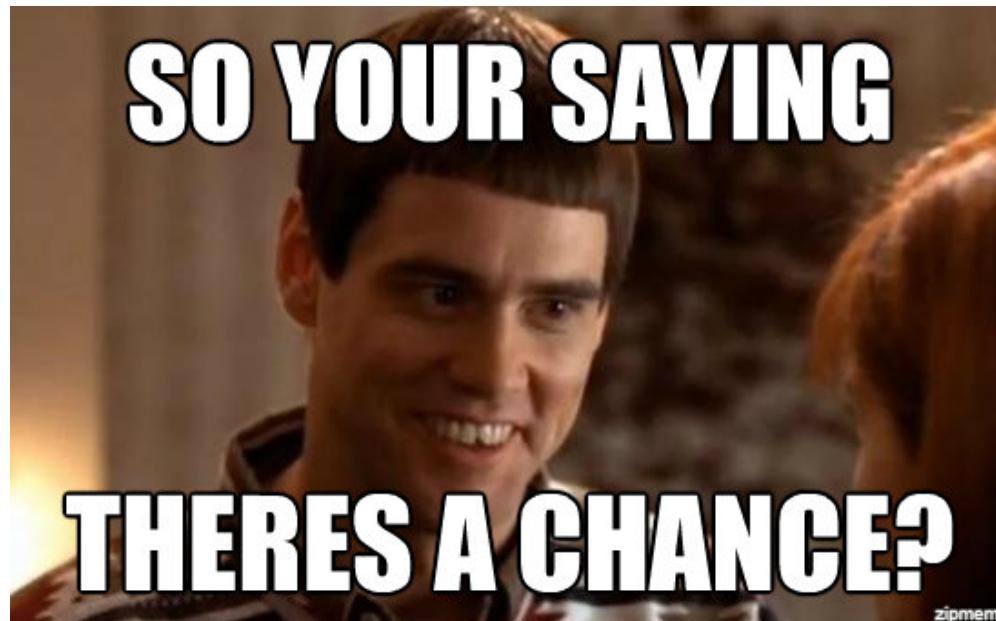
Vamos alinhar as duas
labels nos cantos, usando
estas linhas-guia azuis.





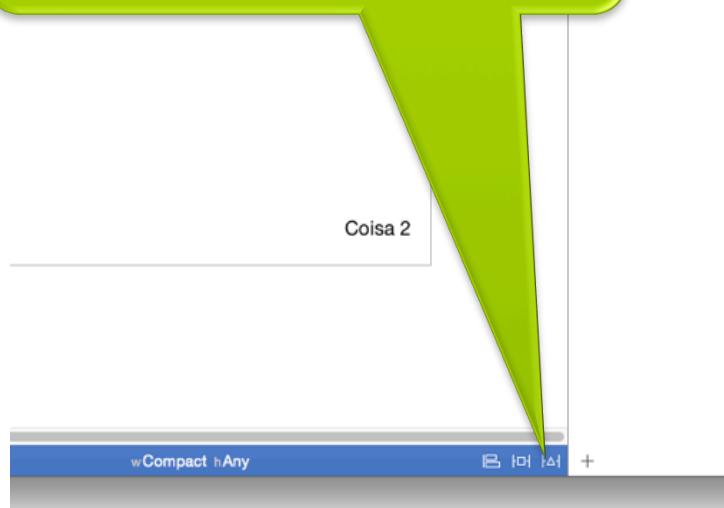
Autolayout

- Felizmente, o Xcode consegue sugerir **constraints**, desde que você use as linhas-guia azuis.

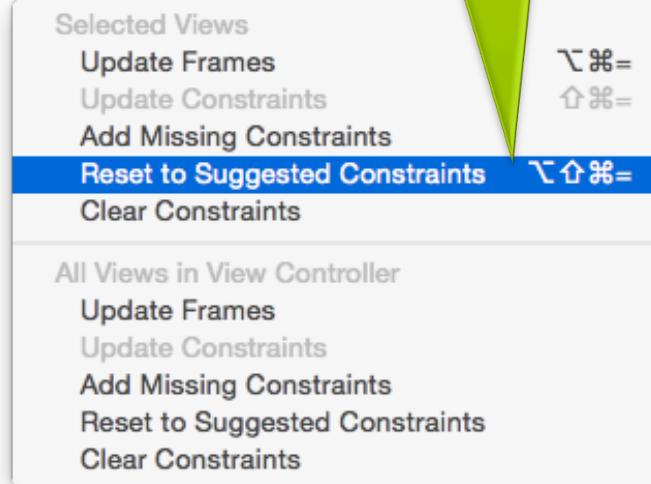


Autolayout

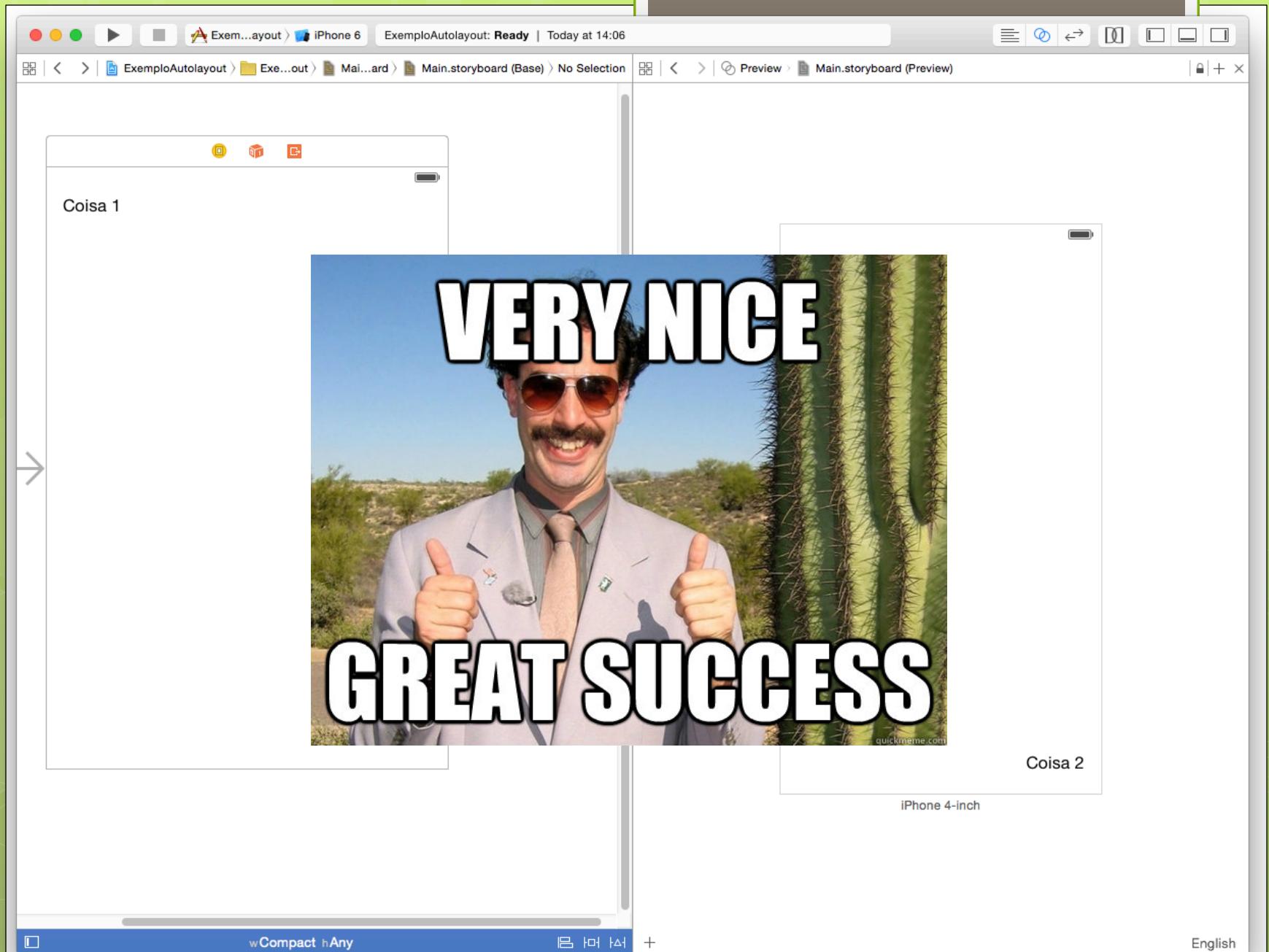
Selecione uma view e
em seguida, clique
neste botão...



E escolha esta opção

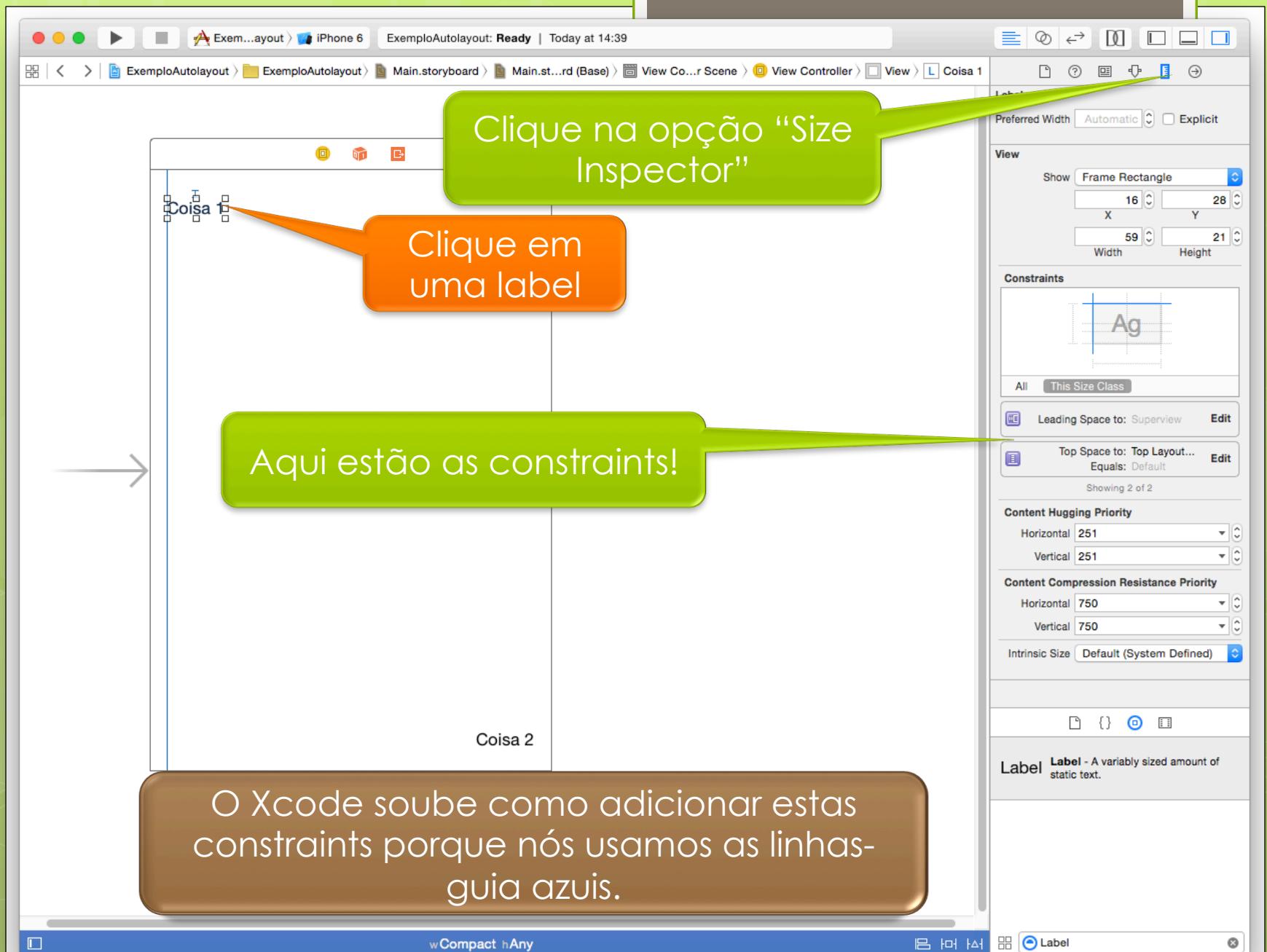


Vamos fazer isso para as duas labels



Autolayout

- Vamos dar uma olhada nas constraints que o Xcode criou para nós...



Autolayout

- Mais na prática:
 - Alinhamento;
 - Como fazer quando não for possível usar as linhas-guia.

UIScrollView

- Serve para mostrar algum conteúdo que seja maior que a tela;
- Oferece rolagem vertical e horizontal e os respectivos efeitos de bounce;
- Oferece funcionalidade de zoom;
- É fácil!

UIScrollView

- Na prática vamos ver:
 1. Como adicionar um conteúdo muito maior do que a tela;
 2. Como controlar, usando o padrão delegate, as ações do scroll view

Multithread (GCD)

- Filas são a essência do multithreading no iOS;
- Blocos podem ser enfileirados (assim como chamadas de métodos);
- Blocos, então, são retirados da fila e executados na thread associada.

Multithread (GCD)

- Main Queue (thread principal)
 - É uma fila especial, onde todas as interações visuais acontecem
- Por convenção, **qualquer coisa que não tenha relação com a tela deve acontecer fora desta fila.**
- Outras Filas
 - Na maior parte das vezes, o iOS vai criar as filas conforme for necessário;
 - Temos também a possibilidade de programaticamente criarmos filas. Vamos dar uma olhada rápida.

Multithread (GCD)

- Executando um bloco em uma outra fila:

Observe que esta não é uma NSString! Trata-se de um array de caracteres da linguagem C (chamado C String)

Função (em C) para criar uma fila

```
dispatch_queue_t minhaFila = dispatch_queue_create("minha fila", NULL);
dispatch_async(minhaFila, ^{
    //Minhas operações aqui.
    //Este código irá rodar em uma outra thread
});
```

Multithread (GCD)

- Como obter a Main Queue:

```
dispatch_queue_t mainQueue = dispatch_get_main_queue();
//ou
NSOperationQueue *mainQ = [NSOperationQueue mainQueue];
```

- Quando usar um ou outro?
 - Veremos em breve.

Multithread (GCD)

- Jeito fácil de chamar um método na thread principal:

```
[self performSelectorOnMainThread:@selector(meuMetodo)
                           withObject:nil
                           waitUntilDone:YES];

//ou

dispatch_async(dispatch_get_main_queue(), ^{
    [self meuMetodo];
});
```

Hora de Brincar!

- Exercitar os conceitos do Autolayout;
- Fazer o download e uma imagem muito grande (tanto em tamanho quanto em bytes) em uma thread separada;
- Exibir a imagem em um UIScrollView, com scroll em todas as direções e zoom;

Desafio

- Implementar o autolayout em todos os desafios anteriores.