



C . e . S . A . R

CENTRO DE ESTUDOS E SISTEMAS AVANÇADOS DO RECIFE

- Rafael Lucena
- 27 anos
- Desde 2013, fazendo história aqui no C.E.S.A.R (todos com iOS)
- 7 anos trabalhando com iOS para diferentes projetos
 - Redes sociais, sistemas de vendas, sistemas de cupons, etc
- Atualmente consultor em soluções iOS da Kozio Digital, Inc. (Chicago)
- Graduado em 2015 no IFPE
- Pós Graduado em 2018 no C.E.S.A.R (Internet das Coisas)



E VOCÊS?



C . E . S . A . R

iOS Avançado

- Layouts usando XIBs separadamente do Storyboard;
- Diagramando telas integradas no mesmo storyboard vs Storyboards/XIBs separados
- Múltiplos Storyboards
- Suportando Layout de tela em comum para diferentes resoluções (aparelhos diferentes)
- Usando telas com ScrollView
- Mudanças de orientações de tela
- **Plus:**
- Compartilhando informações do seu aplicativo (shared)
- Apple Mapkit / Core Location

CONSTRAINTS?

constraints são regras aplicadas às **Views** para que as mesmas se organizem em quais quer proporções que estejam imersas

ou seja

São as regras que fazem os elementos de tela não saírem voando pela tela :)

AUTOLAYOUT

Ferramenta mágica que permite aplicar constraints de maneira muito rápida e visual

Mágica mas que pode ser caótica

TELA DE LOGIN

TELA DE USUARIO LOGADO

BORA COLOCAR A
MÃO NA MASSA?

AGORA VAMOS SEPARAR EM
XIBS DIFERENTES?

- PASSOS:
 - Crie uma segunda view controller
 - Selecione a criação automática de xibs (ou crie o arquivo de xib separadamente)
 - Na sua primeira view controller aplique o seguinte código:

```
let myViewController = MyViewController(nibName: "MyViewController", bundle: nil)
self.present(myViewController, animated: true, completion: nil)
```

Você pode até eliminar todos os storyboards e
usar somente xibs

- PARA ELIMINAR OS STORYBOARDS E USAR SOMENTE OS XIBS
 - Apague todos os storyboards
 - Aplique o seguinte código no app delegate método: `didFinishLaunchingWithOptions`:

```
let myViewController = MyViewController(nibName: "MyViewController", bundle: nil)
self.window?.rootViewController = myViewController
```

TADAAAA!!!

Mas e se eu quiser adicionar
a navegação
(NavigationController)

- Criamos uma navigation controller e inserimos a nossa view controller como **root** e adicionamos como **root** da aplicação

```
let myViewController = AnotherViewController(nibName: "AnotherViewController", bundle: nil)
let navigationController = UINavigationController(rootViewController: myViewController)
self.window?.rootViewController = navigationController
```

```
let myViewController = AnotherViewController(nibName: "AnotherViewController", bundle: nil)
let navigationController = UINavigationController(rootViewController: myViewController)
self.window?.rootViewController = navigationController
```

DESAFIO

Utilize xibs diferentes e navigation controller para criar um sistema de login (com logout no botão de back e mostrando um alerta quando deslogar)

STORYBOARDS

QUAIS AS VANTAGENS DE
USAR UM STORYBOARD?

Tudo em apenas
um lugar

Fluxo de tela
facilmente
compreensível

Muito drag and
drop e menos
código

Bom para
aplicações
básicas

Designers
adoooooram :)

Tudo em apenas
um lugar

Manutenção por
uma equipe

Arquivo de XML
gigante

Merges dignos
de missão
impossível

Relação de
amor e ódio
(talvez mais
ódio)

ENTÃO QUAL A SOLUÇÃO???

Porque não utilizar storyboards separados?

- É comum utilizar Storyboards separados para manter a coesão das telas
- Cada storyboard representa uma feature da aplicação
 - Podemos ter como exemplo um storyboard que seja responsável pelo onboarding da aplicação por exemplo
- Storyboards separados devem ser auto-suficientes
 - Devem apenas receber, apresentar e/ou retornar dados e só!!!
 - Essa separação deve ser utilizada para fechar um pacote de features que pode ser usada em qualquer ocasião dentro da aplicação

VAMOS SEPARAR EM STORYBOARDS VAMOS?

Vamos criar um Onboarding

Com uma foto de vocês :)

AGORA VAMOS USAR OS DOIS AO MESMO TEMPO

Vamos criar dois estados para nosso
Onboarding

Com outra foto de vocês :)

VAMOS TROCAR NOSSO
TARGET PARA UM IPHONE SE



C . E . S . A . R

**Diferentes
resoluções**

- Devemos sempre nos preocupar com a resolução em outros aparelhos
- No mundo iOS temos poucos tamanhos de tela para nos preocupar (comparados aos 12374691723 milhões de modelos de Android que existem por aí)

iPhone SE

iPhone XR

iPhone 7/8

iPhone XS

iPhone 7/8 plus

iPhone XS Max



E TOME CONSTRAINTS

Constraints fixas

X

Constraints dinâmicas

- Constraints padrão
 - Leading
 - Trailing
 - Top
 - Bottom
 - Height
 - Width
 - Aspect Ratio
 - Y alignment
 - X alignment
- Constraints específicas:
 - UIImageView: content mode
 - UILabel: text alignment, line break

- Constraints proporcionais à largura e/ou altura
- Constraints proporcionais aos eixos (vertical, horizontal)
- Constraints com valor máximo e mínimo
- Constraints de alinhamento à outras view
 - top, bottom, leading, trailing, center X and center Y
- Constraints hugging (problema comum de largura e altura)
 - exemplo: duas views que não possuem largura e se alinham horizontalmente uma à outra

AGORA VAMOS ADICIONAR
SUPORTE AOS IPHONES SE/
5/5S

AGORA VAMOS ADICIONAR
SUPORTE AOS IPHONES X /
XR / XS

MAS E SE A GENTE
ROTACIONAR A TELA?





C . E . S . A . R

Orientações de tela

- Duas soluções:
 - Criar telas proporcionais o suficiente (layouts mais básicos)
 - Gerenciar a troca de constraints manualmente (diretamente na view controller)
 - Criar outra view controller e aplicar as constraints nela (comum em telas que diferem muito entre si)

```
override func willTransition(to newCollection: UITraitCollection,  
with coordinator: UIViewControllerTransitionCoordinator) {
```

AGORA VAMOS ORGANIZAR
TODA A NOSSA APLICAÇÃO
PARA LANDSCAPE E PORTRAIT

DESAFIO

Crie um Todo App usando xibs diferentes em conjunto com storyboards, que tenha um Onboarding que explique a aplicação e que funcione em todas as orientações

- Todo App:
 - Xibs diferentes
 - Storyboards diferentes
 - Suporte à todas as orientações
 - Table View
 - Collection View no Onboarding (ponto extra)
- Funcionalidades:
 - Tela de Splash
 - Onboarding (collection view)
 - *sem suporte a landscape
 - Tela de Todos (table view)
 - *sem suporte a landscape
 - CRUD de Todos (utilizar UserDefaults para persistir os dados)
 - Tela de adição/edição/visualização de Todos (utilizando a mesma tela)
 - Remoção de Todos



C . e . S . A . R

MapKit

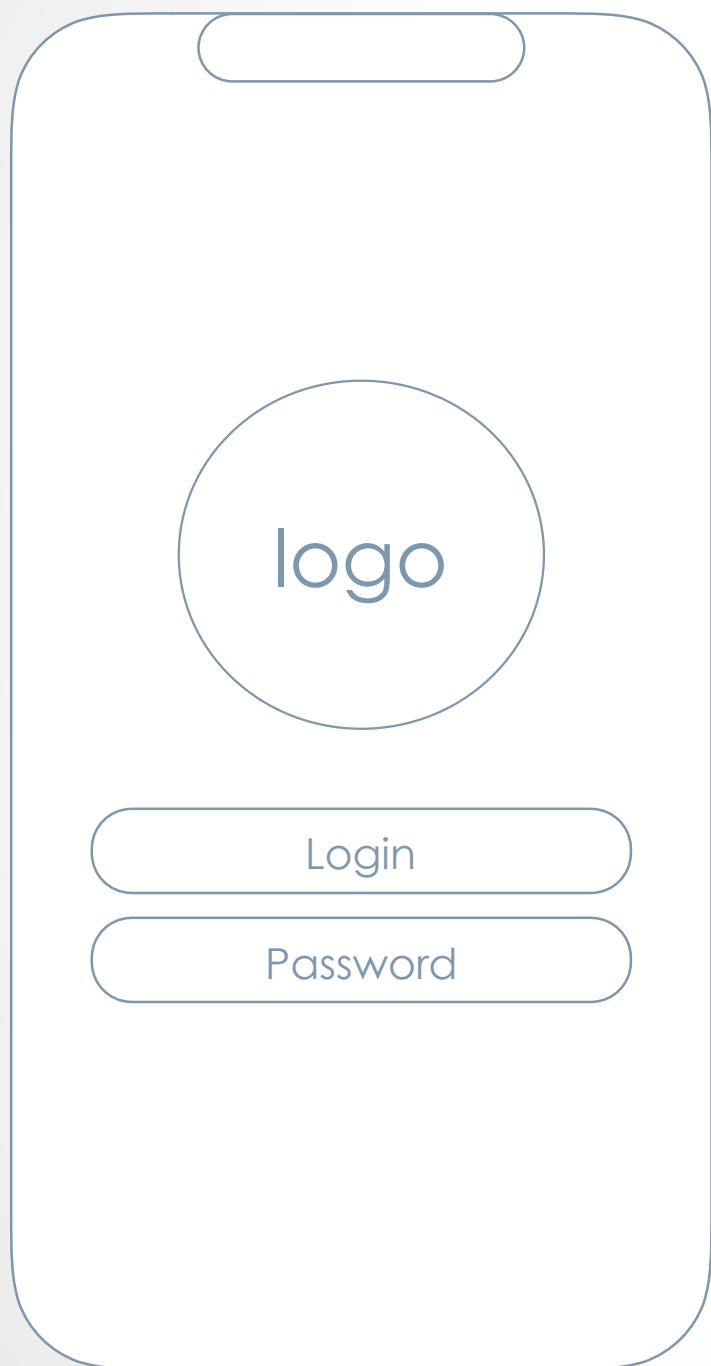
- O MapKit é uma api para apresentação de mapas usando o Maps imbutido no iOS
 - *mapkit não funciona juntamente com o google maps
- Para adicionar um mapa em sua aplicação é necessário implementar os seguintes métodos:
 - adicione uma map view na sua view controller
 - link suas constraints e pronto!!

- Para solucionar o problema de zoom e localização do usuário:
 - método para checar se os serviços estão ativos
 - checar permissões
 - adicionar no info.plist
 - criar método para checar as permissões do usuário
- implementar o LocationManager (CoreLocation)
 - locationManager:didUpdateLocations
 - locationManager:didChangeAuthorization
- para colocar a localização atual:
 - `mapView.showsUserLocation = true`

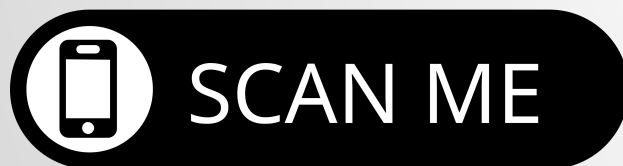
AGORA ADICIONAR
ANOTAÇÕES NO MAPA

E AGORA PRA FINALIZAR,
VAMOS TERMINAR O NOSSO
PROJETO E ADICIONAR
SOMENTE UMA COISINHA LÁ :)

- Games App:
 - Xibs diferentes
 - Storyboards diferentes
 - Suporte à todas as orientações
 - Table View
- Funcionalidades:
 - Onboarding (collection view)
 - *sem suporte a landscape
 - Tela de login
 - *com suporte a landscape
 - Tela de Games (table view)
 - *sem suporte a landscape
 - Tela de Games (collection view)
 - *com suporte a landscape
 - a collection view deve ser vertical
 - Tela de adição de jogos
 - Ao adicionar novos jogos, atualizar a table view e collection view automaticamente (não atualizar a table view e collection view no viewWillAppear)
 - Atualizar automaticamente utilizando UINotification



- Rafael Lucena
- Email do C.E.S.A.R: rlmg@cesar.org.br
- Email pessoal: rafellk@gmail.com



LinkedIn



Twitter



Instagram





C . e . S . A . R

CENTRO DE ESTUDOS E SISTEMAS AVANÇADOS DO RECIFE