Com quantas views se faz uma APP?

Auto Layout com Layout Anchor, Stack Views e Criatividade

Agenda

Storyboards? XIBs? View Code?

Layout Anchor

Stack Views

Bônus

Conclusão

Storyboards? XIBs? View Code?

```
H:|-[left]-50-[right]-|
NSLayoutContraint
```

Layout Anchor

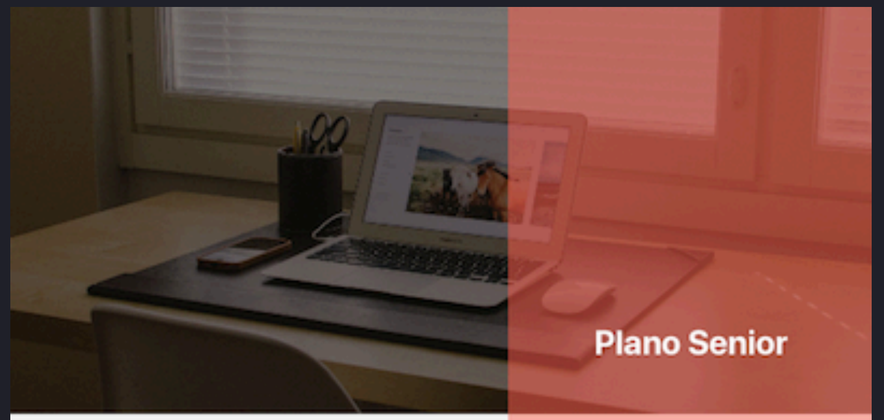




NSLayoutContraint

“Ainda que a classe **`NSLayoutAnchor`** tenha a checagem adicional de tipo, ainda é possível criar constraints inválidas. Por exemplo, o compilador permite que você configure a constraint **`leadingAnchor`** de uma view com a **`leftAnchor`** de outra view, pois ambas são instâncias de **`NSLayoutXAxisAnchor`**. Entretanto, Auto Layout não permite constraints que misturem atributos de **`leading`** e **`trailing`** com atributos de **`left`** ou **`right`**. Como resultado, as constraints vão quebrar em tempo de execução.”

– Documentação da Apple



Plano Senior

Nome Sobrenome
Desenvolvedor de Software
(11) 1111-1111
nome@sobrenome.com.br

Nome Sobrenome
Desenvolvedor de Software
(11) 1111-1111
nome@sobrenome.com.br

Plano Pleno

Nome Sobrenome
Desenvolvedor de Software
nome@sobrenome.com.br
(11) 1111-1111

Plano Junior

Layout Anchor

```
final class BusinessCardComponent {
    let photoImageView: UIImageView
    let nameLabel: UILabel
    let titleLabel: UILabel
    let phoneLabel: UILabel
    let emailLabel: UILabel

    // ...

}
```

Layout Anchor

```
final class SeniorBusinessCardView: UIView {

    private let businessCardComponents: BusinessCardComponents

    init(businessCardComponents: BusinessCardComponents) {
        self.businessCardComponents = businessCardComponents
        super.init(frame: .zero)
        setupViewHierarchy()
        setupConstraints()
    }

    private func setupViewHierarchy() {
        addSubview(businessCardComponents.photoImageView)
    }

    private func setupConstraints() {
        businessCardComponents.photoImageView.topAnchor.constraint(equalTo: topAnchor).isActive = true
        businessCardComponents.photoImageView.leadingAnchor.constraint(equalTo: leadingAnchor).isActive = true
        businessCardComponents.photoImageView.trailingAnchor.constraint(equalTo: trailingAnchor).isActive = true
        businessCardComponents.photoImageView.heightAnchor.constraint(equalToConstant: imageHeight).isActive = true

        // ...

    }

}
```

Layout Anchor

```
extension UIView {

    @discardableResult func topAnchor(equalTo anchor: NSLayoutYAxisAnchor, constant: CGFloat = 0) -> Self {
        topAnchor.constraint(equalTo: anchor, constant: constant).isActive = true
        return self
    }

    @discardableResult func bottomAnchor(equalTo anchor: NSLayoutYAxisAnchor, constant: CGFloat = 0) -> Self {
        bottomAnchor.constraint(equalTo: anchor, constant: constant).isActive = true
        return self
    }

    // ...

}
```

Layout Anchor

```
final class SeniorBusinessCardView: UIView {

    // ...

    private func setupConstraints() {
        businessCardComponents.photoImageView
            .topAnchor(equalTo: topAnchor)
            .leadingAnchor(equalTo: leadingAnchor)
            .trailingAnchor(equalTo: trailingAnchor)
            .heightAnchor(equalTo: imageHeight)
    }

}
```

Stack Views

“A stack view está de acordo com Auto Layout (é baseada no layout de constraints do sistema) para organizar e alinhar uma lista de views de acordo com sua especificação. Para tirar proveito da stack view, você precisa saber o básico sobre as contraints do Auto Layout, como descrito no Guia do Auto Layout”

– Documentação da Apple

Stack Views

```
final class JuniorBusinessCardView: UIView {
    // ...
    private let stackContentContainer: UIStackView = {
        let stack = UIStackView(frame: .zero)
        stack.axis = .vertical
        stack.translatesAutoresizingMaskIntoConstraints = false
        return stack
    }()
    init(businessCardComponents: BusinessCardComponents) {
        self.businessCardComponents = businessCardComponents
        super.init(frame: .zero)
        setupViewHierarchy()
        setupConstraints()
    }
    private func setupViewHierarchy() {
        stackContentContainer.addArrangedSubview(businessCardComponents.nameLabel)
        // ...
    }
    private func setupConstraints() {
        // ...
        stackContentContainer
            .topAnchor(equalTo: businessCardComponents.photoImageView.topAnchor)
            .leadingAnchor(equalTo: businessCardComponents.photoImageView.trailingAnchor, constant: defaultMargin)
            .trailingAnchor(equalTo: trailingAnchor, constant: defaultMargin)
            .bottomAnchor(equalTo: businessCardComponents.photoImageView.bottomAnchor)
    }
}
```

Bônus

Conclusão

👋 👍 ☝ ?

. . . . . .

- http://equinocios.com/view-code/2017/03/18/com-quantas-views-se-faz-um-APP/
- https://developer.apple.com/reference/uikit/nslayoutanchor
- https://developer.apple.com/library/etc/redirect/xcode/content/1189/documentation/UserExperience/Conceptual/AutolayoutPG/index.html#//apple_ref/doc/uid/TP40010853
- https://developer.apple.com/reference/appkit/nsstackview

- CocoaHeads Slack: ronan
- ronanrodrigo.com