

# Reutilizar código

*Baby steps para hacer un  
framework*



# SwiftBA



---

# Daniela P. Riesgo

Ing. en Informática, FIUBA

Desarrolladora iOS  
desde 2016



**SwiftBA**

# Reutilizar código

# Hacer un framework

# Reutilizar código

## ¿Para qué?

- Desarrollo más rápido
- Evitar bugs

## ¿Cómo?

- Funciones
- Templates
- Componentes
- Módulos
- Librerías
- Frameworks

# Hacer un framework

# Reutilizar código

## ¿Para qué?

- Desarrollo más rápido
- Evitar bugs

## ¿Cómo?

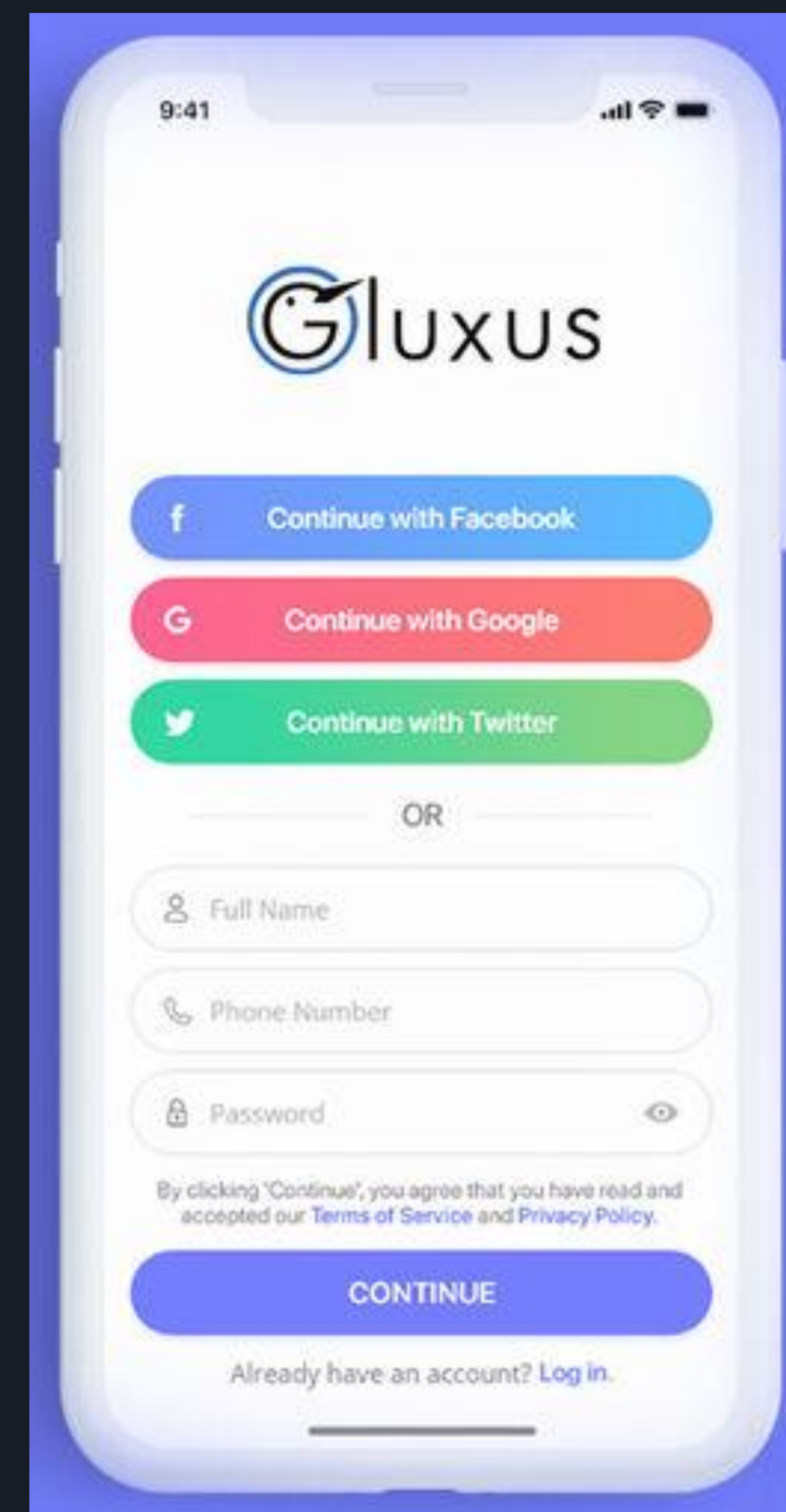
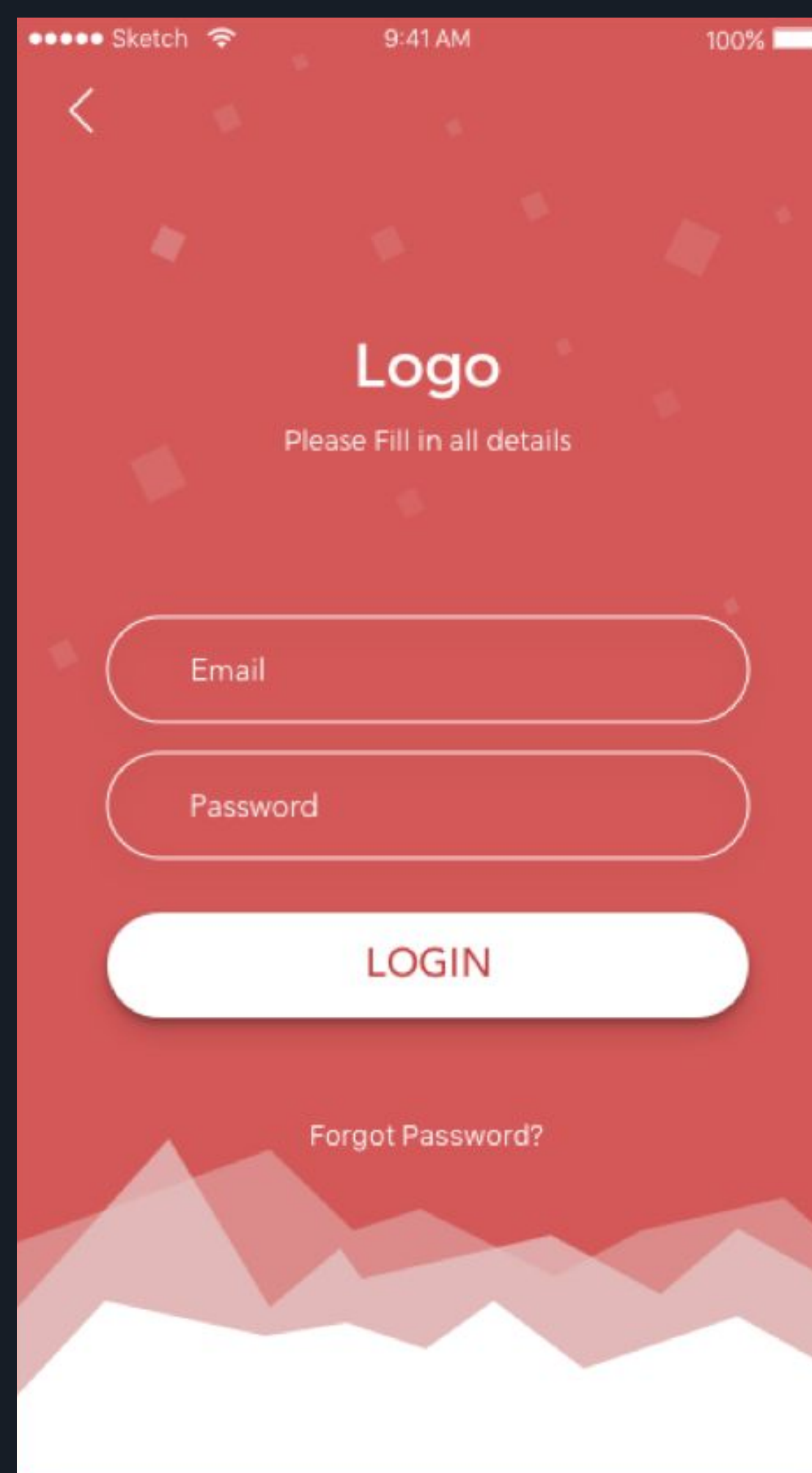
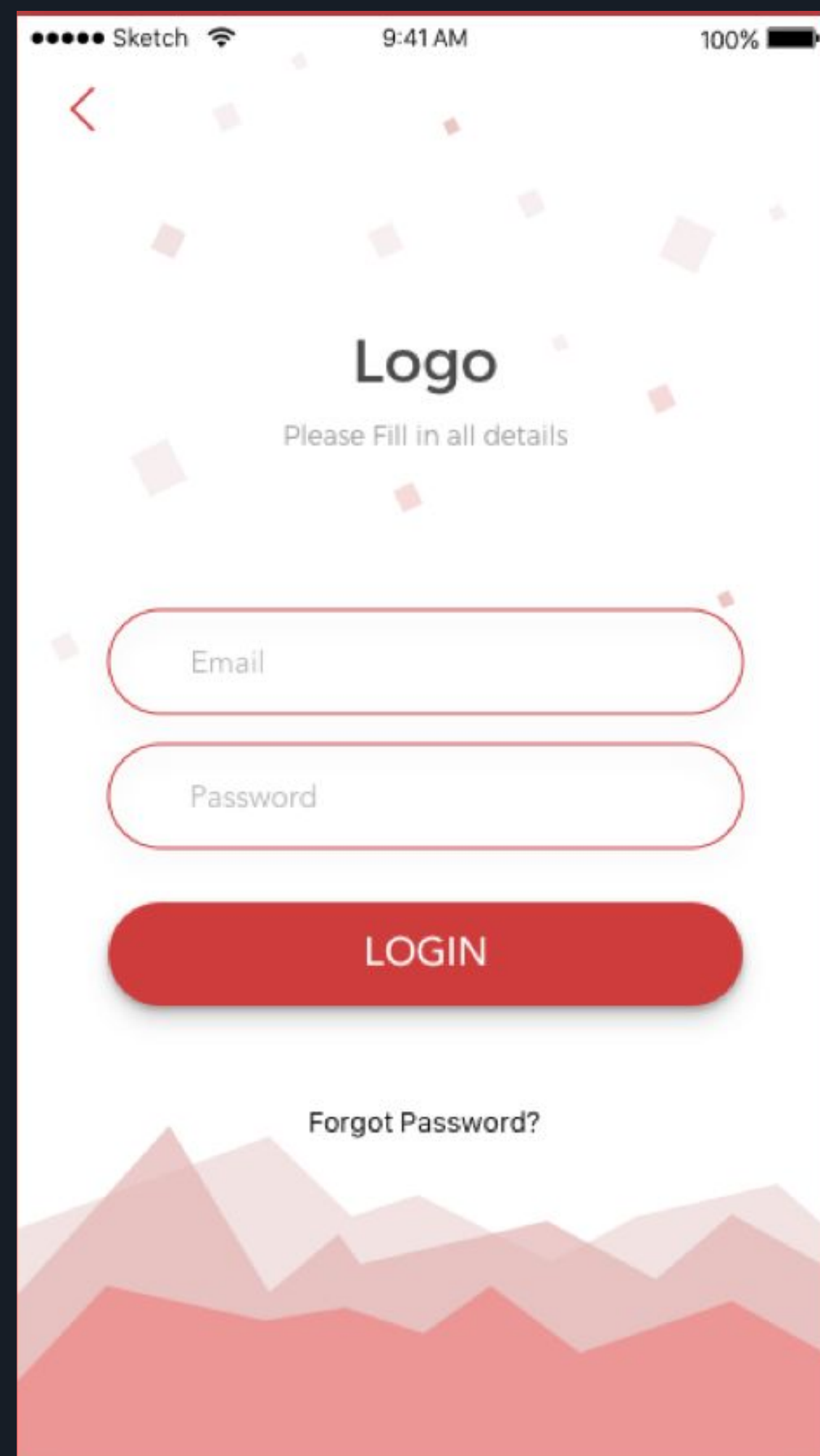
- Funciones
- Templates
- Componentes
- Módulos
- Librerías
- Frameworks

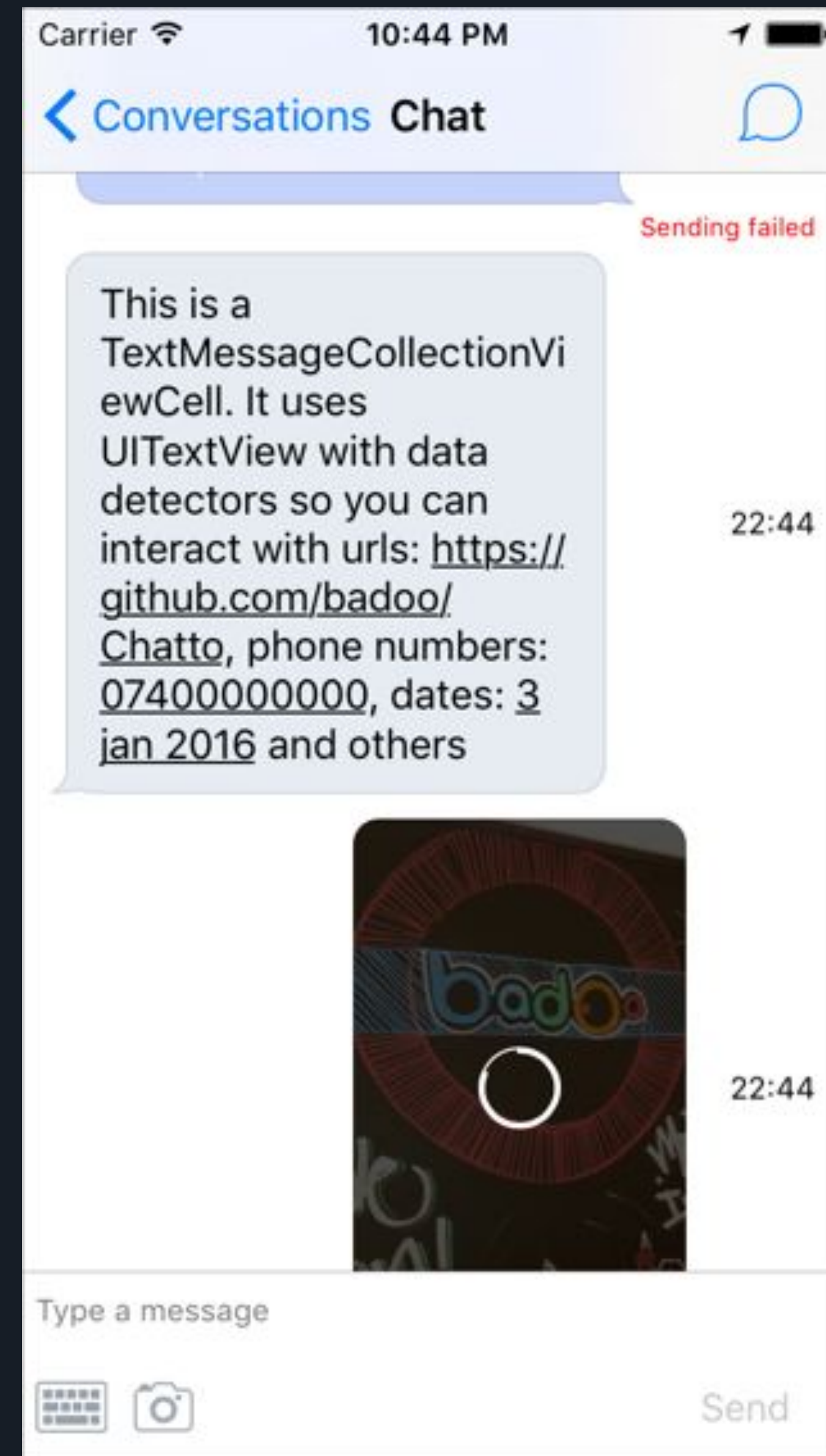
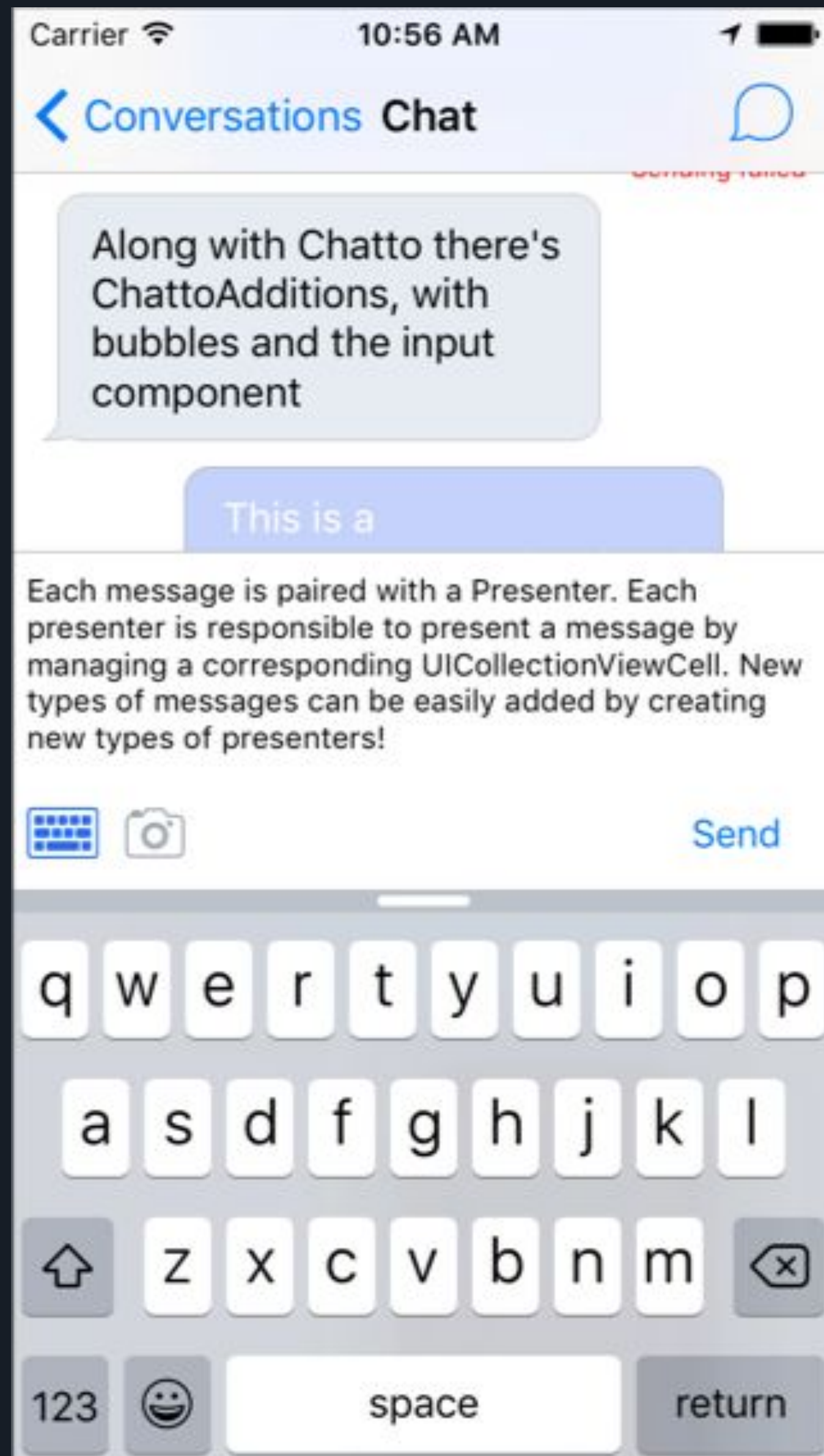
# Hacer un framework

## Tipos

- Core (para desarrollo o arquitectura)
- Adicionales de desarrollo
- Feature SDK
- Enlatados





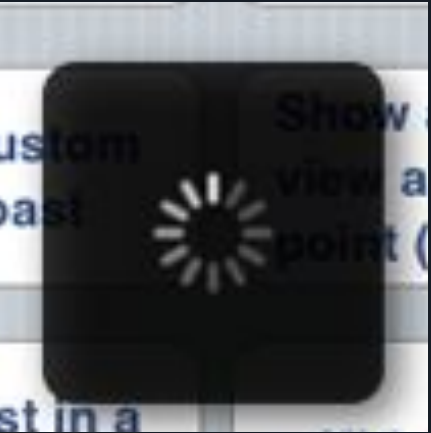




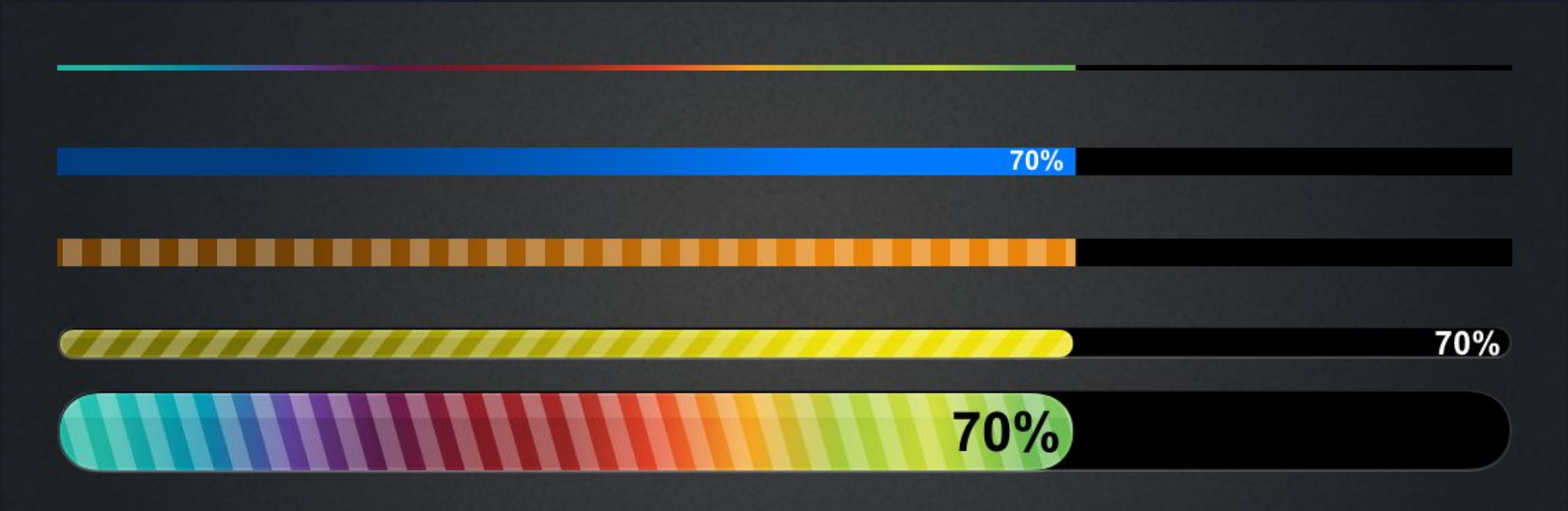
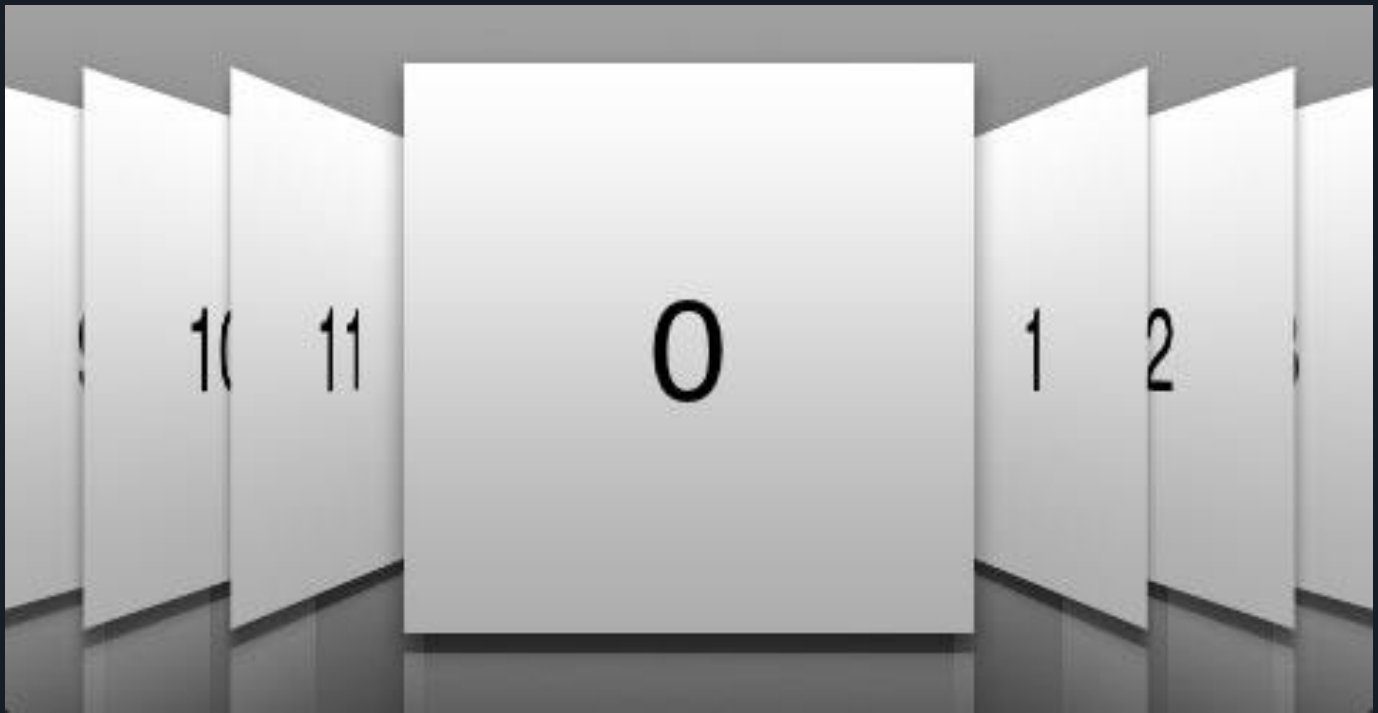
hello@hi.

Password

| M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|
|    | 1  | 2  | 3  | 4  | 5  | 6  |
| 7  | 8  | 9  | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 |    |    |    |    |



This is a piece of toast.





```
extension NibLoadable where Self: UIView {

    static func loadFromNib(_ nibName: String? = nil) -> Self? {
        let bundle = Bundle(for: self)
        let probableName = "\(self)".split { $0 == "." }.map(String.init).last!
        let name = nibName ?? probableName
        let nib = UINib(nibName: name, bundle: bundle)
        guard let view = nib.instantiate(withOwner: self, options: nil).first as? Self else {
            return .none
        }
        return view
    }
}
```

```
class AudioPlayer: NSObject, AudioPlayable {  
  
    internal let audioPlayer: AVAudioPlayer  
  
    public protocol AudioSessionManageable {  
        func startAudioSession()  
    }  
  
    public class AudioSessionManager: AudioSessionManageable {
```

```
class KeyboardedViewController: UIViewController {  
  
    @IBOutlet weak var bottomContainerConstraint: NSLayoutConstraint?  
    private let margin: CGFloat = -8.0  
    private var notificactionInfo: NSNotification?
```

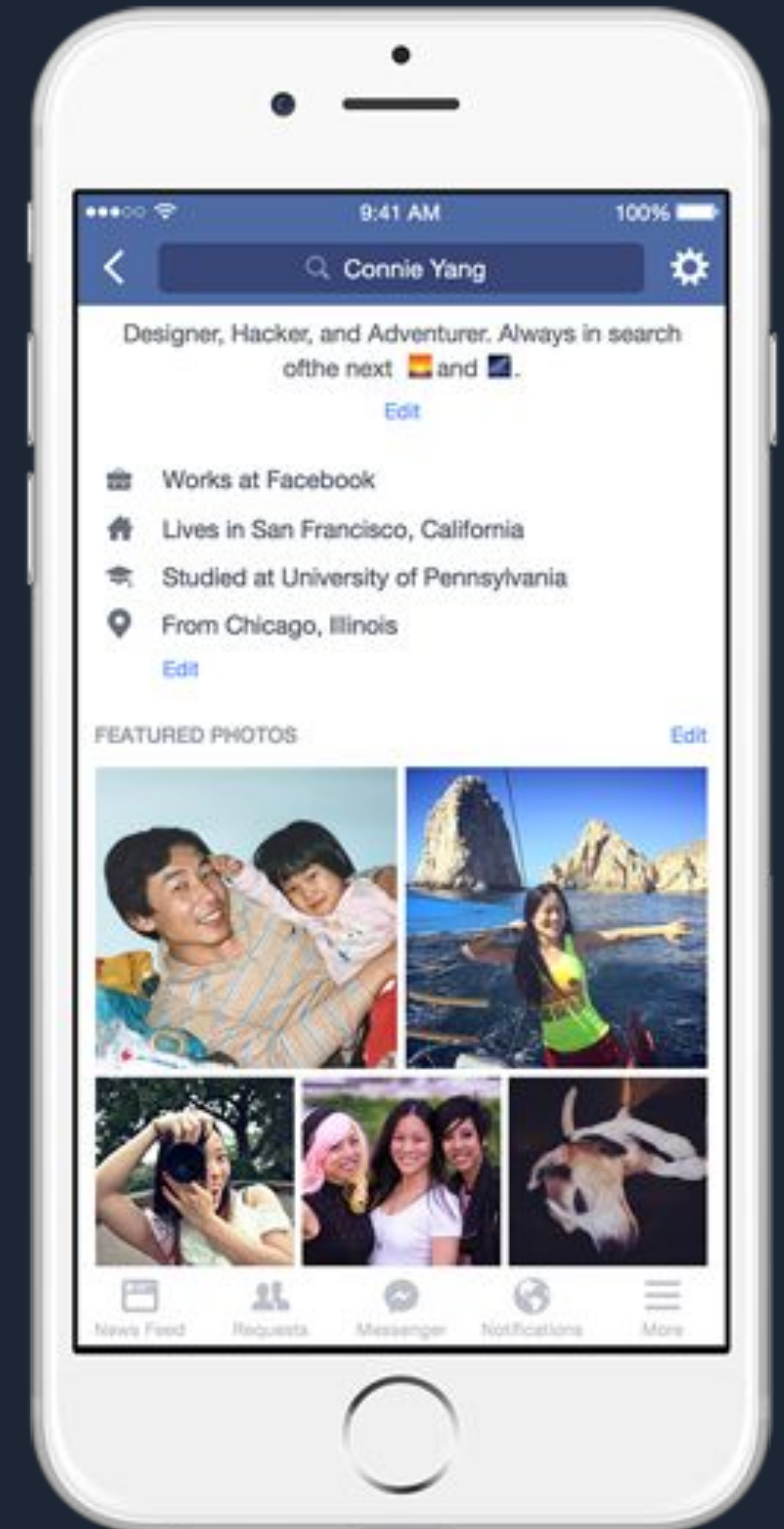
```
class PopUpViewController<ViewModel>: UIViewController where ViewModel: PopUpViewModelable {  
  
    @IBOutlet weak var animationView: UIView!  
    @IBOutlet private weak var mainIcon: UIImageView!
```



# Mentalidad generalista

**Pensemos en cómo conseguimos esas primeras 5 fotos.**

**Supongamos que si el usuario A tiene bloqueado al usuario B y el usuario B entra al perfil de un usuario C, no debe ver ninguna foto de A.**





# Mentalidad generalista

Pensemos en cómo conseguimos esas primeras 5 fotos.

Supongamos que si el usuario A tiene bloqueado al usuario B y el usuario B entra al perfil de un usuario C, no debe ver ninguna foto de A.

for  
filter && prefix



# Mentalidad generalista

Pensemos en cómo conseguimos esas primeras 5 fotos.

Supongamos que si el usuario A tiene bloqueado al usuario B y el usuario B entra al perfil de un usuario C, no debe ver ninguna foto de A.

for  
filter && prefix

first(n, where:)



**Cosas a reusar hay muchas**

**Conocemos los beneficios de reusarlas y no rehacerlas  
o copiarlas**



**Cosas a reusar hay muchas**

**Conocemos los beneficios de reusarlas y no rehacerlas  
o copiarlas**

**¡Hagamos un  
framework!**

**Cosas a reusar hay muchas**

**Conocemos los beneficios de reusarlas y no rehacerlas  
o copiarlas**

**¡Hagamos un  
framework!**

**... pero  
¿cómo y de  
qué tipo?**

**1.**

**Encontrarlo**

**2.**

**Integrarlo**

**3.**

**Usarlo**



# 1. Buscar, encontrar y decidir probarlo

1. Tener público el repo
2. Tener un buen README con una buena licencia
3. Tener actividad reciente: issues y projects/boards de Github/Gitlab
4. Tener soporte para varias plataformas
5. Tener forks y estrellas
6. ¿Cuántas dependencias me agrega?  
¿Alguna colisiona con las mías?
7. ¿Cuán customizable es?
8. ¿Cuán fácil es hacer lo que quiero?
9. Tiene versiones compatibles con mi proyecto

## 2. Integrarlo

1. Demo que logre hacer correr rápidamente
2. Disponible en mi dependency manager de preferencia
3. Instrucciones de instalación
4. Scripts o comandos para automatizar lo pesado
5. Que lo pueda agregar en mi proyecto para algo básico con el más mínimo setup de código

# 3. Usarlo

1. Customizarlo para lo que quiero
2. Leer más sobre cómo funciona
3. Subir issues con preguntas/problemas
4. Fork con cambios y PR
5. Copiar y pegar el código y cambiarlo



# El camino al framework

*Mitos, verdades y consejos*



SwiftBA

**WolmoCore**  
**WolmoUtils**

**WolmoAuth**  
**WolmoNet**

**Enlatado**

Changelog

Versiones de  
Xcode y Swift

Retrocompatibilidad

Actualizar

**Versionado**

Releases

Reducirlo

Tests

CI

Dependency managers

Uso

Nombrado

Desarrollo

Simplificar

**Customización**

Documentación

Dependencias

Demo



Calidad

Soporte

Actualizaciones

**Customización**

Compleitud

Compatibilidad

¿y entonces?



1.

Hacer un framework es *mucho* trabajo

2.

**Hay que lograr agregar valor, *fácil y rápido***

3.

O hacer algo muy *complejo* la mitad de  
difícil y lento



4.

**Hasta el MVP y más allá**

¿Preguntas?



¡GRACIAS!

