

TipKit

Sugiriendo funcionalidades a tus usuarios

¡Hola! 🖐️

Soy Misael Cuevas 🇩🇴



💻 iOS Software Engineer @ Kindred Group plc

✍️ Escribo artículos sobre iOS @ asynclearn.com

🤠 Me gusta el wrestling, One Piece, Overwatch y Zelda

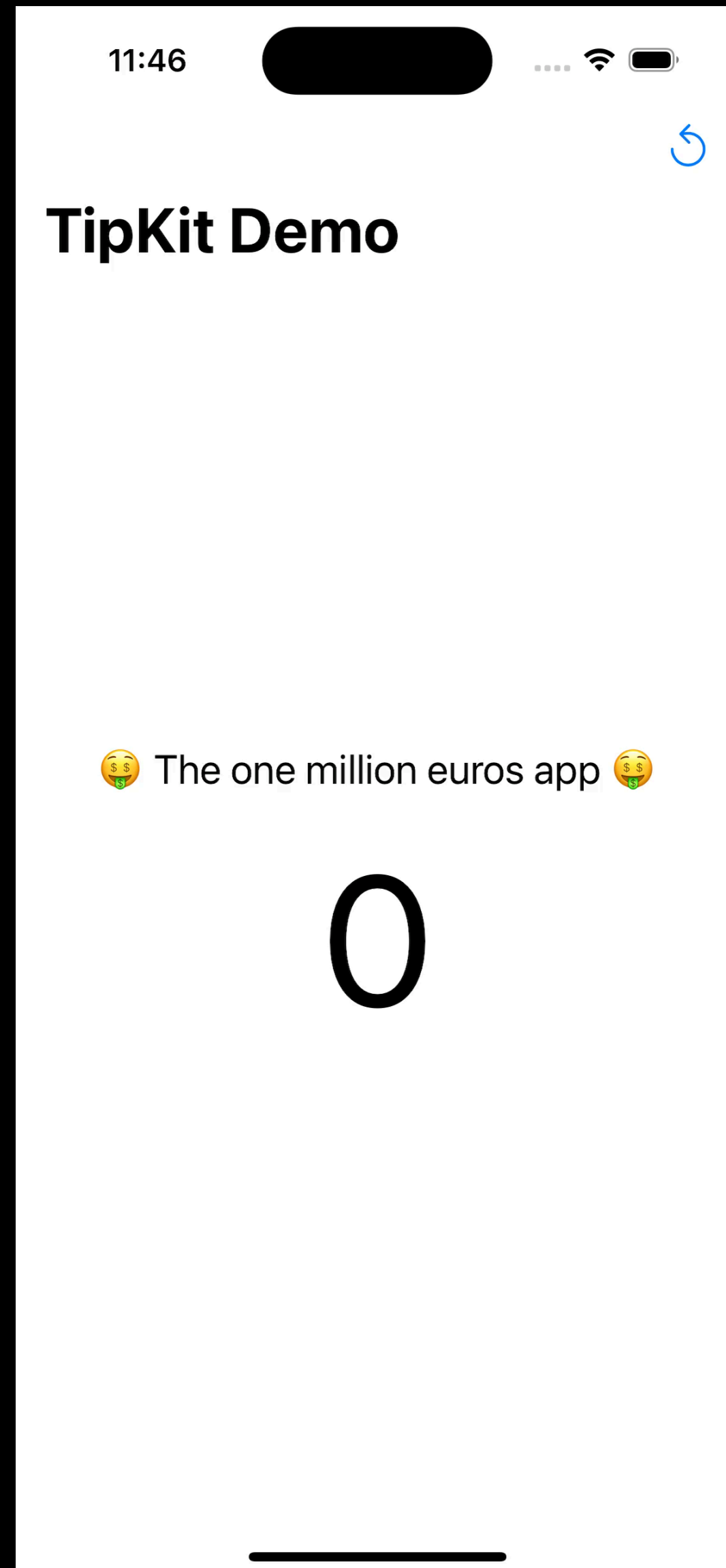
X @mcvasquez09

in @mcvasquez

Agenda

- 1 Crear un Tip.
- 2 Agregar acciones.
- 3 Personalizar estilos.
- 4 Definir reglas.
- 5 Otras configuraciones.
- 6 Testing.

App Demo



TipKit

iOS 17.0+

iPadOS 17.0+

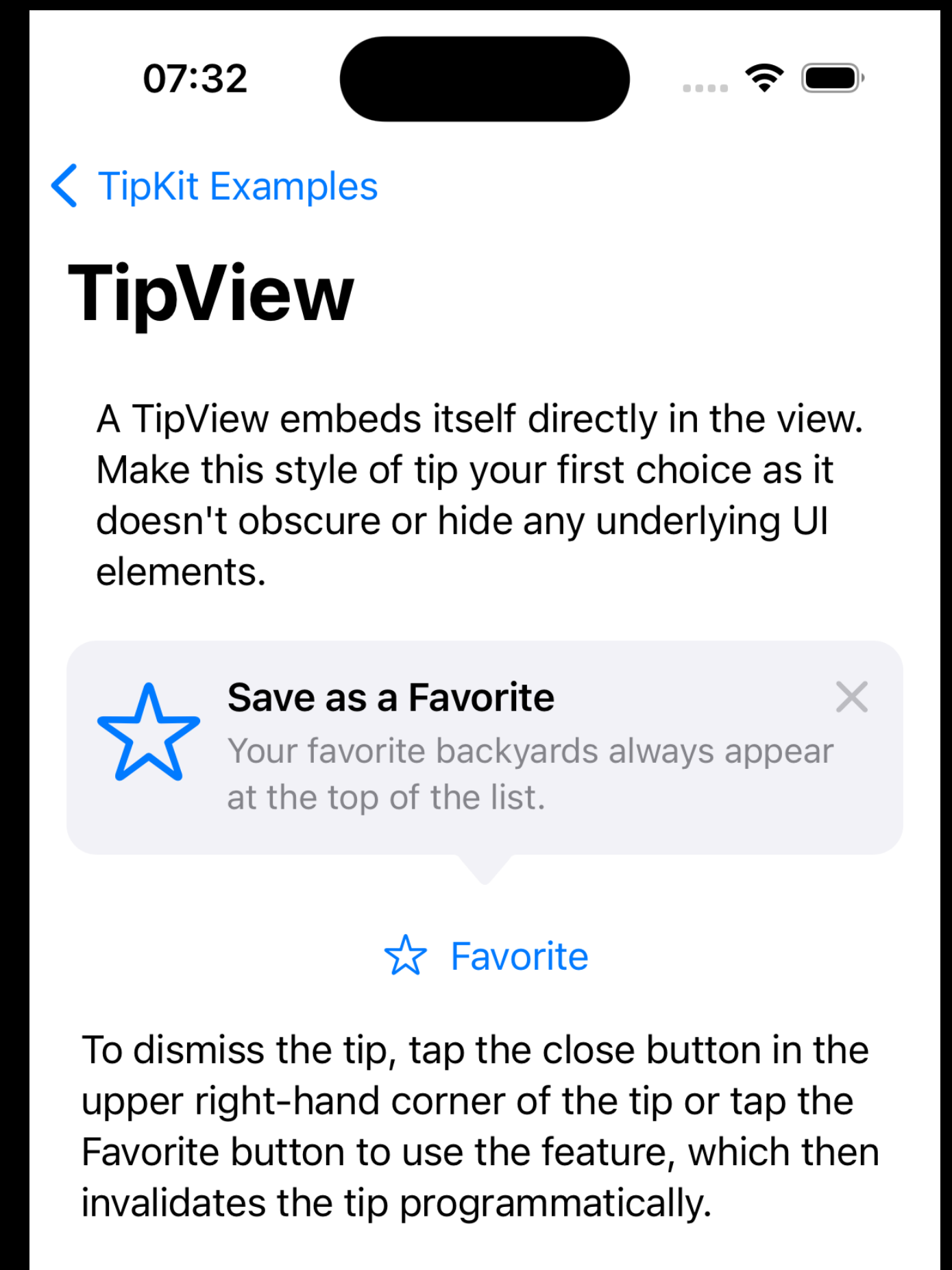
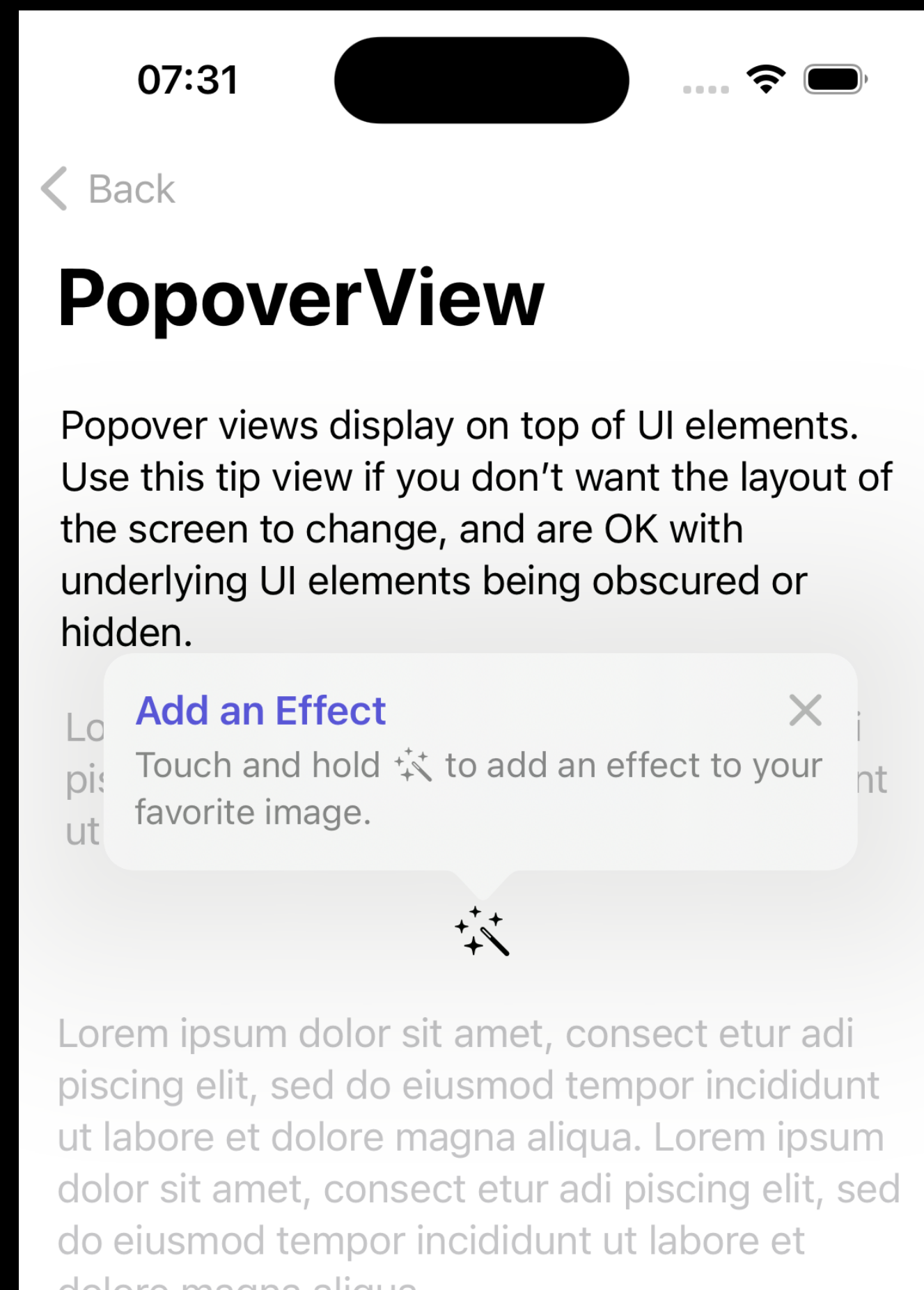
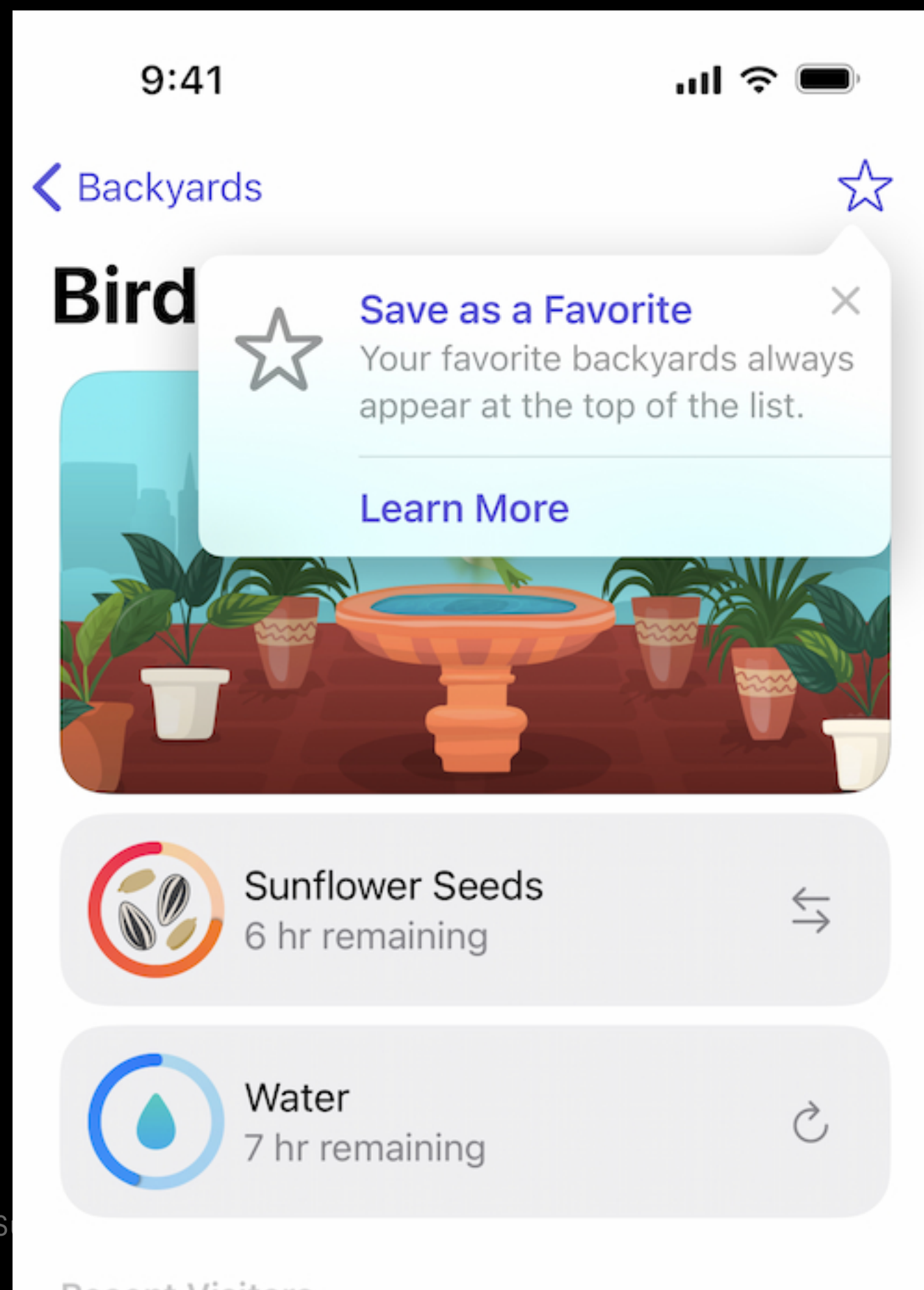
macOS 14.0+

Mac Catalyst 17.0+

tvOS 17.0+

watchOS 10.0+

visionOS 1.0+



Habilitar TipKit

1

```
import SwiftUI

@main
struct TipKitDemoApp: App {
    var body: some Scene {
        WindowGroup {
            ContentView()
        }
    }
}
```

1

```
import SwiftUI
import TipKit

@main
struct TipKitDemoApp: App {
    var body: some Scene {
        WindowGroup {
            ContentView()
                .task { try? Tips.configure() }
        }
    }
}
```


2

```
import SwiftUI

@main
struct TipKitDemoApp: App {
    var body: some Scene {
        WindowGroup {
            ContentView()
        }
    }
}
```

2

```
import SwiftUI
import TipKit

@main
struct TipKitDemoApp: App {
    init() { try? Tips.configure() }

    var body: some Scene {
        WindowGroup {
            ContentView()
        }
    }
}
```

Crear un Tip

```
import TipKit

struct MyTip: Tip {
    var title: Text {
        Text("Mi primer tip 😍")
    }
}
```

```
import TipKit

struct MyTip: Tip {
    var title: Text {
        Text("Mi primer tip 🥰")
    }

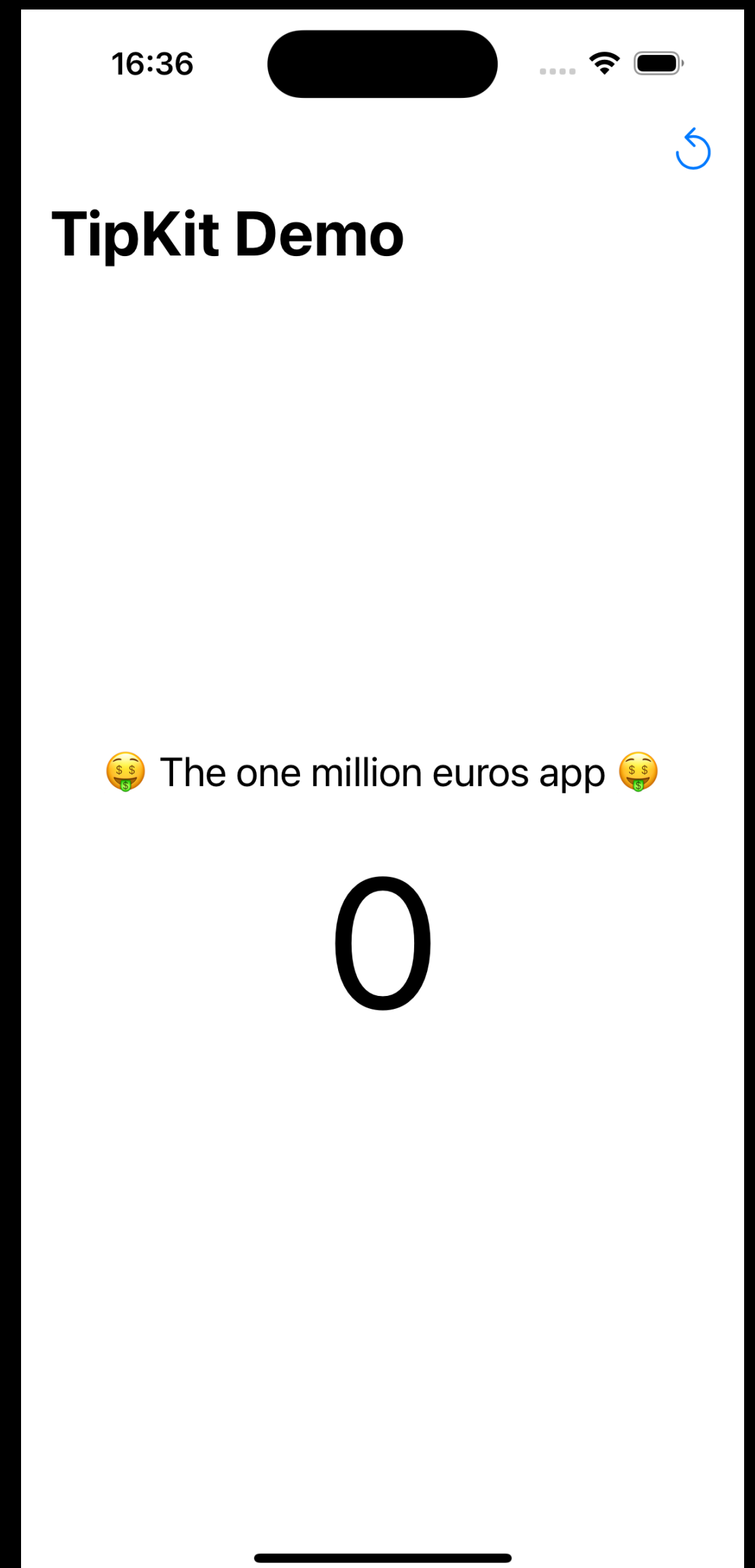
    var message: Text? {
        Text("Un gran poder conlleva una gran responsabilidad.")
    }

    var image: Image? {
        Image(systemName: "hand.thumbsup.fill")
    }
}
```


Usar tip en una vista de SwiftUI

```
import SwiftUI
```

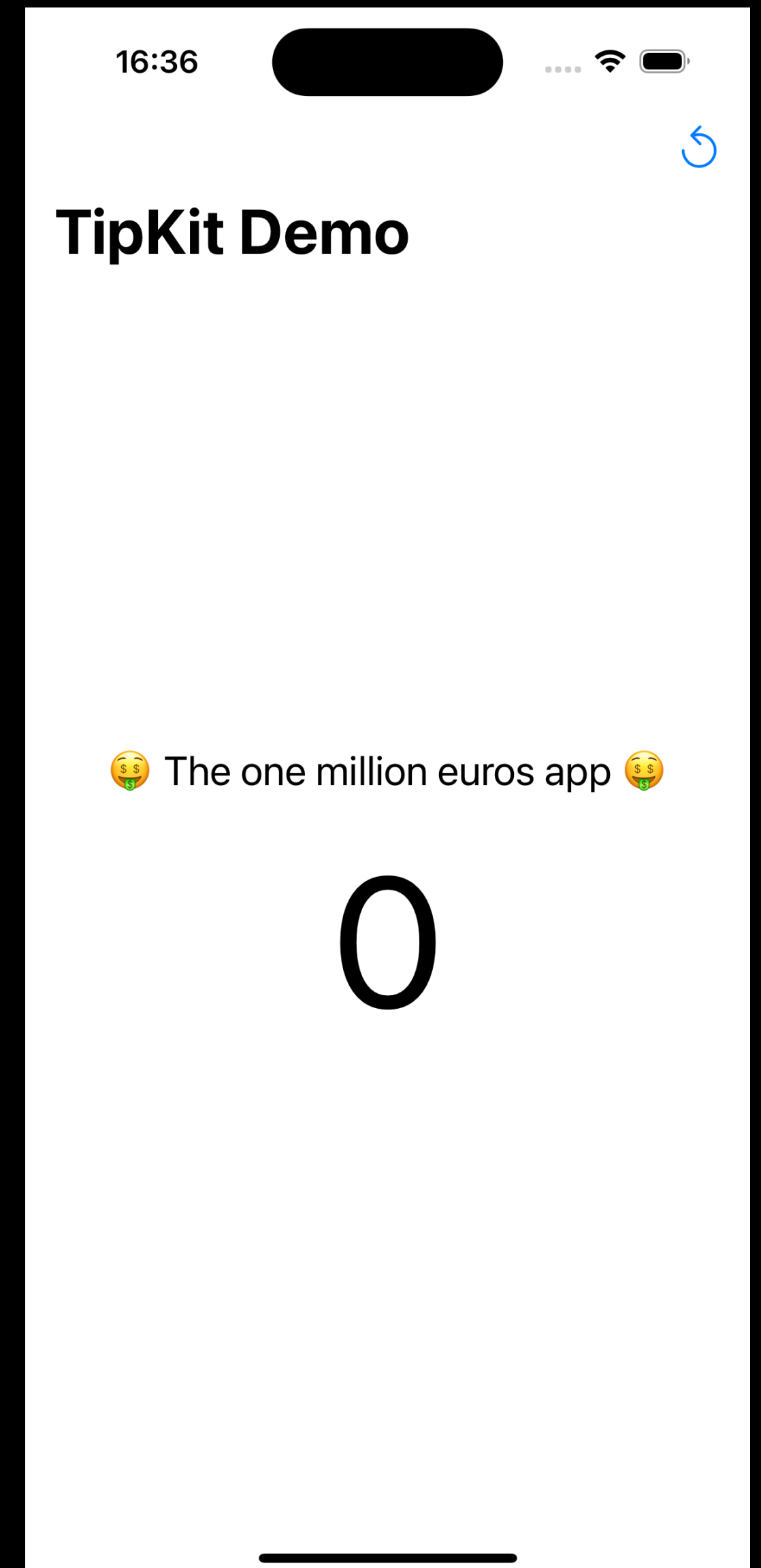
```
struct ContentView: View {  
    var body: some View {  
        ...  
    }  
}
```



```
import SwiftUI
import TipKit

struct ContentView: View {
    var myTip = MyTip()

    var body: some View {
        ...
    }
}
```

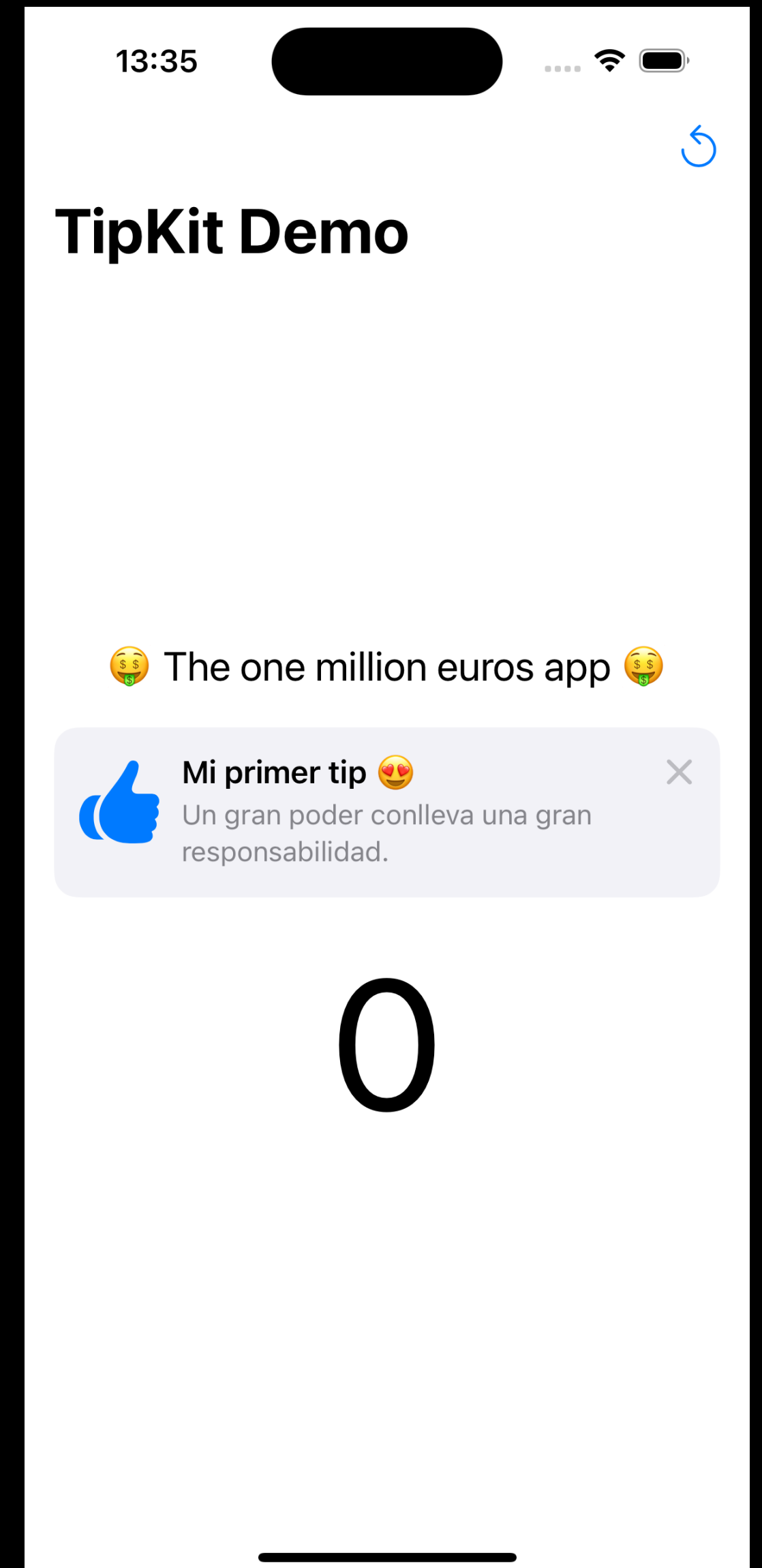


Inline Tip

```
import SwiftUI
import TipKit

struct ContentView: View {
    var myTip = MyTip()

    var body: some View {
        ...
        TipView(myTip)
        ...
    }
}
```

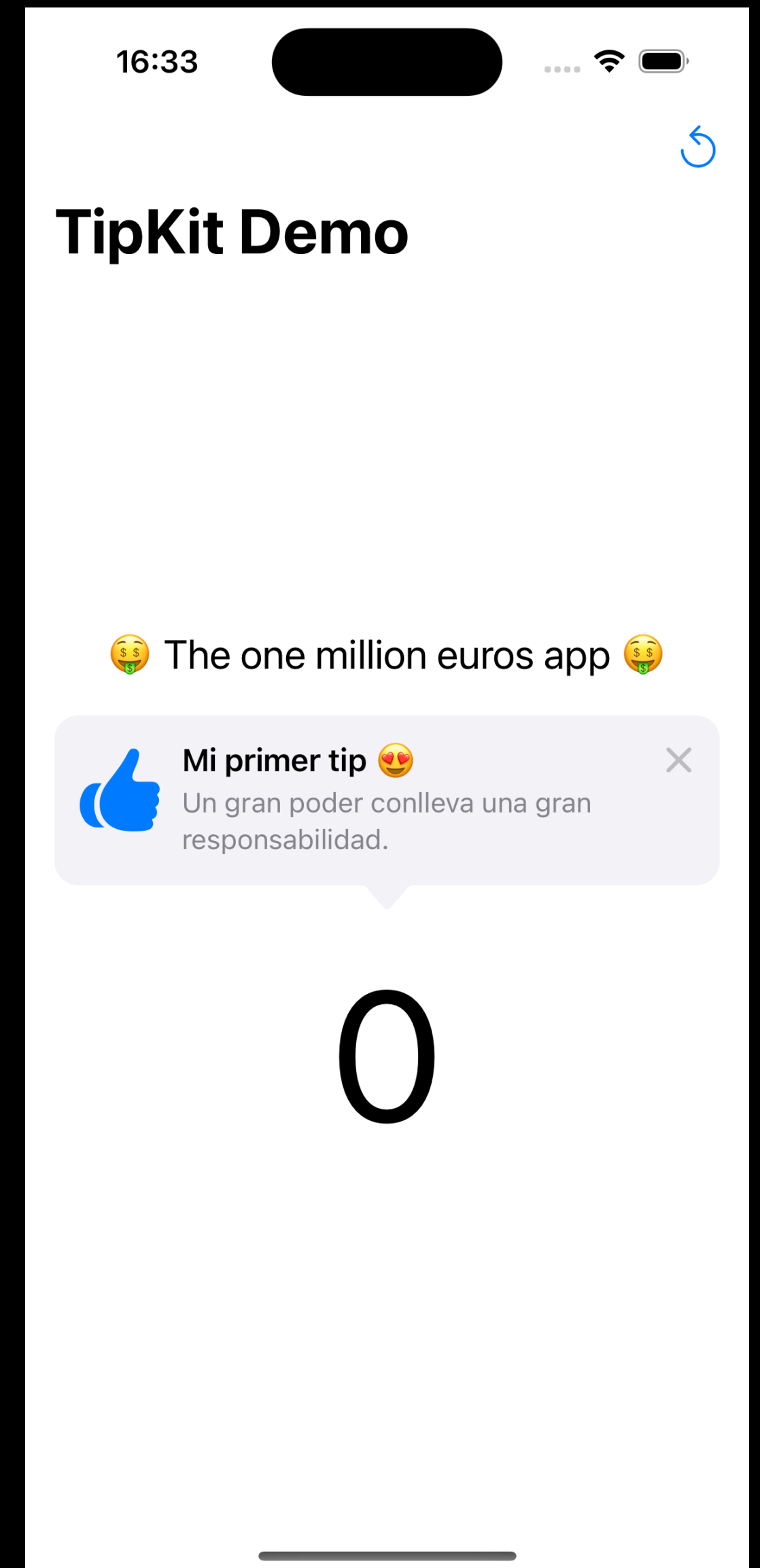


Inline Tip

```
import SwiftUI
import TipKit

struct ContentView: View {
    var myTip = MyTip()

    var body: some View {
        ...
        TipView(myTip, arrowEdge: .bottom)
        ...
    }
}
```

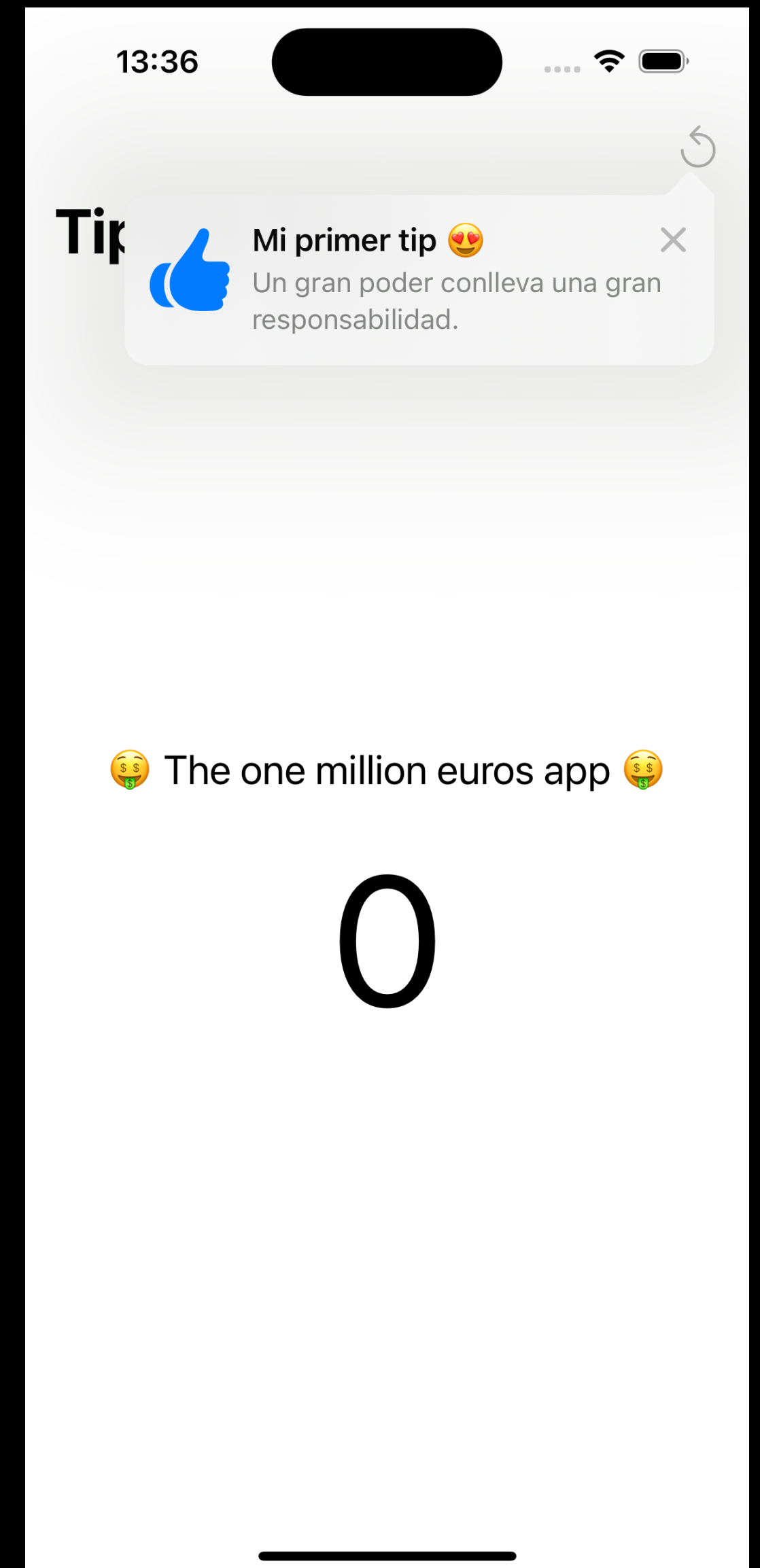


Popover Tip

```
import SwiftUI
import TipKit

struct ContentView: View {
    var myTip = MyTip()

    var body: some View {
        ...
        Button("", systemImage: "...") { }
            .buttonStyle(.borderless)
            .popoverTip(myTip)
    }
}
```



¿Cómo cerrar un tip?

```
myTip.invalidate(reason: InvalidationReason)
```

Razones para invalidar un tip:

- 1 `actionPerformed`
- 2 `displayCountExceeded`
- 3 `tipClosed`

Ejemplo:

```
myTip.invalidate(reason: .actionPerformed)
```

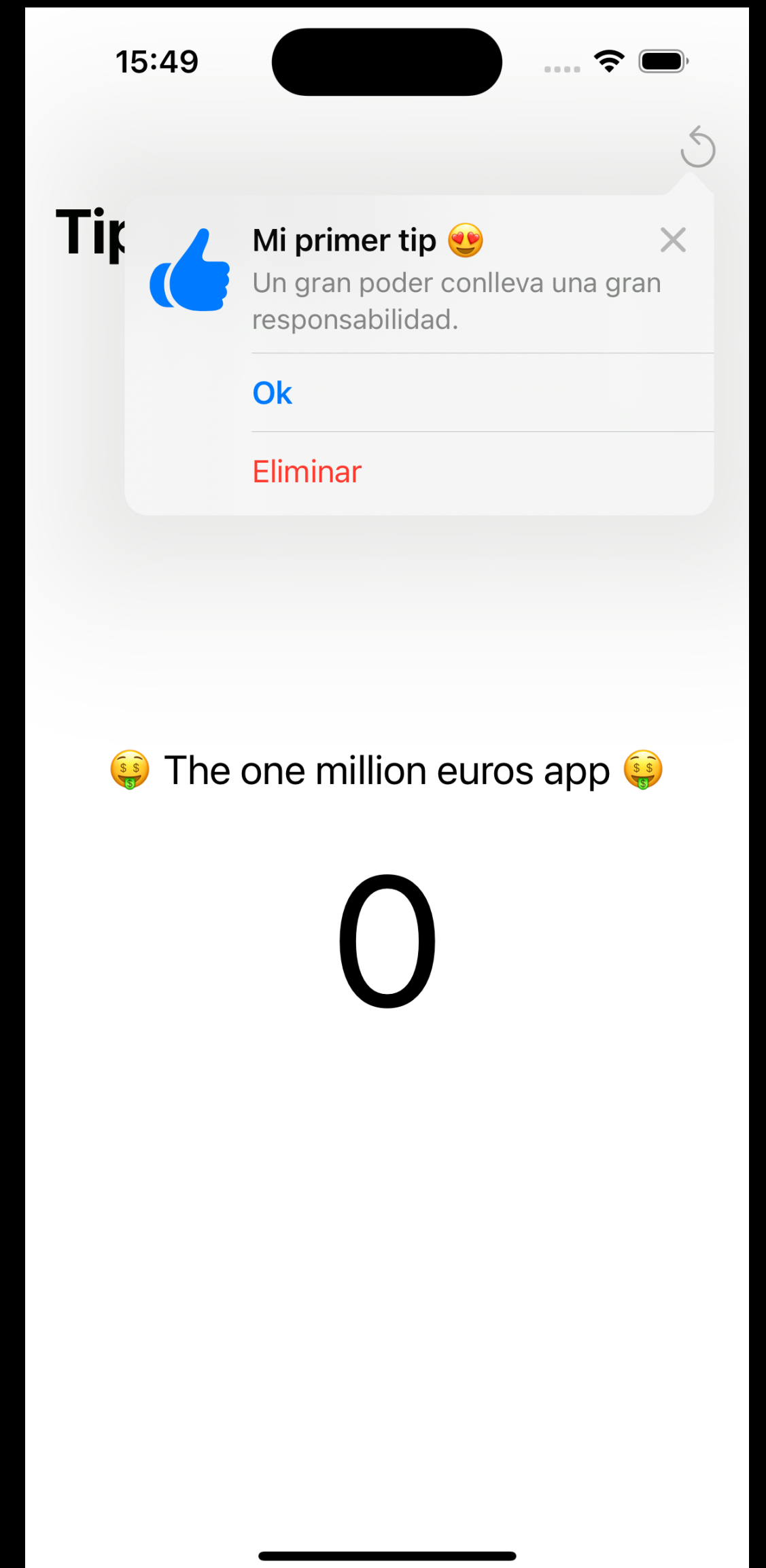
Agregar acciones

```

import TipKit

struct MyTip: Tip {
    ...
    var actions: [Action] {
        [
            Action(title: "Ok"),
            Action {
                Text("Eliminar")
                    .foregroundColor(.red)
            }
        ]
    }
}

```



Inicializadores de Action

```
Action(  
  id: String?,  
  perform: () -> Void,  
  label: () -> Text  
)
```

```
Action(  
  id: String,  
  title: StringProtocol,  
  perform: () -> Void  
)
```


¿Cómo detectar la acción desde la View?

```
TipView(myTip) { action in
    if action.id == "action-ok" {
        ...
    }
}
```

```
Button("", systemImage: "arrow.counterclockwise") { }
    .popoverTip(keynoteTip) { action in
        if action.id == "action-ok" {
            ...
        }
    }
}
```

🧐 ¡Live Demo! 🧐

Personalizar estilos de un Tip

Tip

```
import TipKit

struct MyTip: Tip {
    var title: Text {
        Text("Mi primer tip 😍")
            .bold()
            .foregroundColor(.green)
    }
    var message: Text? {
        Text("Un gran poder conlleva una gran responsabilidad.")
            .font(.caption)
    }
}
```

Mi primer tip 😍

Un gran poder conlleva una gran responsabilidad.



ViewModifiers

Establecer un corner radius:

```
func tipCornerRadius(  
    _ cornerRadius: CGFloat,  
    antialiased: Bool = true  
) → some View
```

Ejemplo:

```
TipView(myTip)  
    .tipCornerRadius(2)
```

Mi primer tip 🥰

Un gran poder conlleva una gran
responsabilidad.



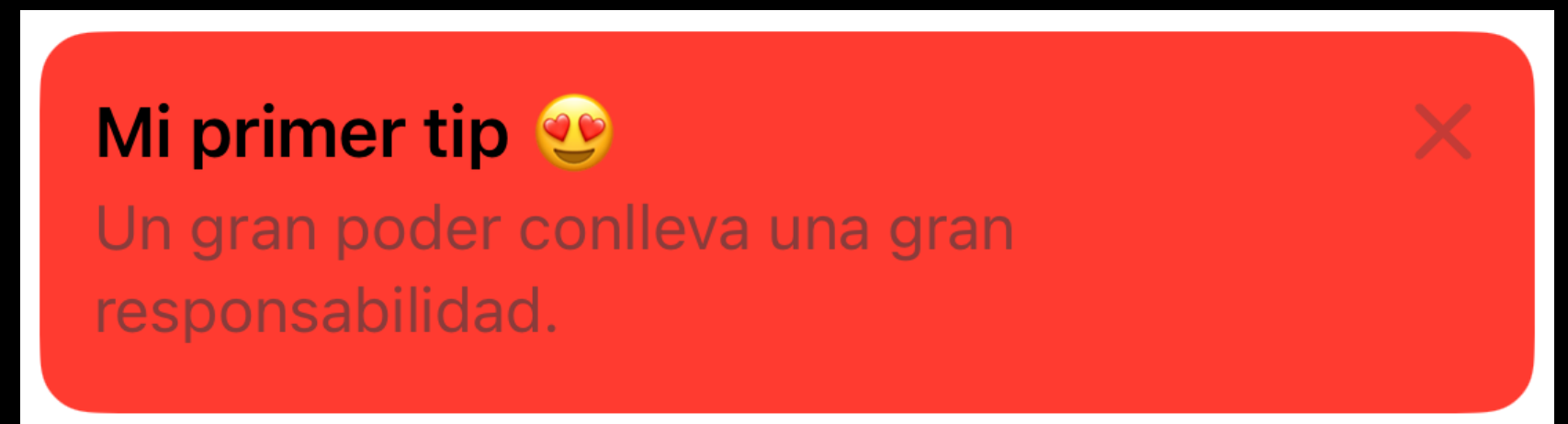
ViewModifiers

Establecer un background al tip (sólo aplica a Inline tips):

```
func tipBackground(_ style: some ShapeStyle) → some View
```

Ejemplo:

```
TipView(myTip)  
    .tipBackground(.red)
```



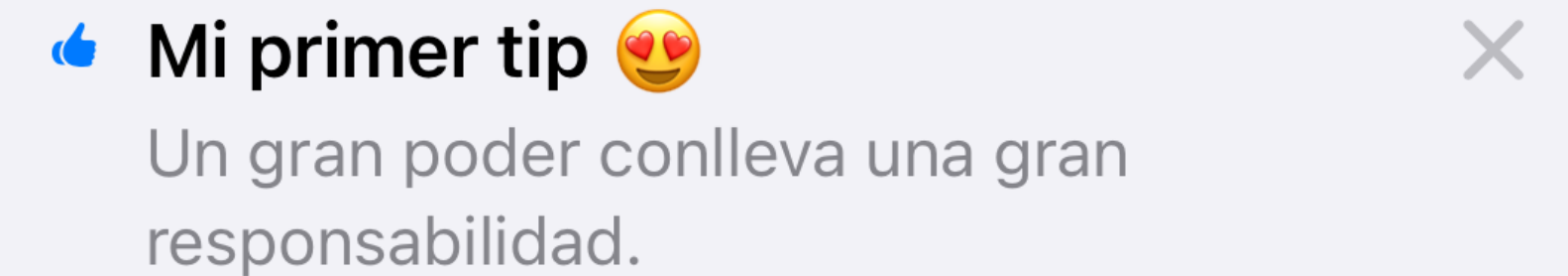
ViewModifiers

Establecer el tamaño de la imagen:

```
func tipImageSize(_ size: CGSize) → some View
```

Ejemplo:

```
TipView(myTip)  
    .tipImageSize(.init(width: 10, height: 10))
```



👍 Mi primer tip 🥰
Un gran poder conlleva una gran responsabilidad.

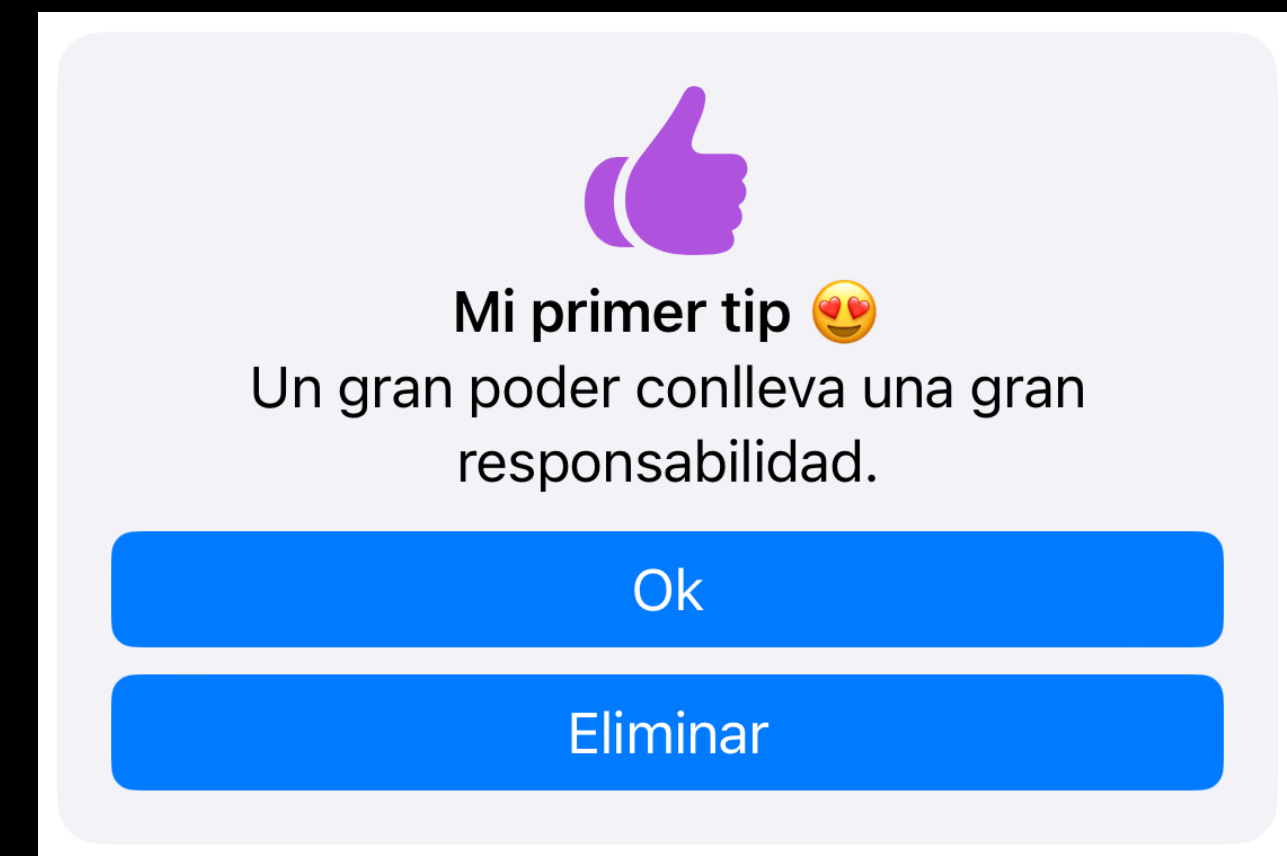
ViewModifiers

Establecer un view style:

```
func tipViewStyle(_ style: some TipViewStyle) → some View
```

Ejemplo:

```
TipView(myTip)  
    .tipViewStyle(MyTipViewStyle())
```



Definir Reglas

Parámetros

```
import TipKit

struct MyTip: Tip {
    ...
    @Parameter
    static var isCounterTapped: Bool = false
}
```

Parámetros

```
import TipKit

struct MyTip: Tip {
    ...
    @Parameter
    static var isCounterTapped: Bool = false

    var rules: [Rule] {
        [
            #Rule(Self.$isCounterTapped) { $0 == true }
        ]
    }
}
```

Parámetros

Ejemplo de cómo usarlo:

```
MyTip.isCounterTapped.toggle()
```


Parámetros

Utiliza **transient** para volver a su valor por defecto en cada sesión de la app. Ejemplo:

```
import TipKit

struct MyTip: Tip {
    ...
    @Parameter(.transient)
    static var isCounterTapped: Bool = false
}
```

Eventos

```
import TipKit

struct MyTip: Tip {
    ...
    static let didTriggerTapTextEvent = Event(id: "event")
}
```

Eventos

```
import TipKit

struct MyTip: Tip {
    ...
    static let didTriggerTapTextEvent = Event(id: "event")

    var rules: [Rule] {
        [
            #Rule(Self.didTriggerTapTextEvent) {
                $0.donations.count > 2
            }
        ]
    }
}
```

Eventos

Ejemplo de cómo usarlo de forma sincrónica:

```
MyTip.didTriggerTapTextEvent.sendDonation()
```

Ejemplo de cómo usarlo de forma asíncrona:

```
await MyTip.didTriggerTapTextEvent.donate()
```

Eventos

Puedes filtrar las donaciones a partir de un rango de tiempo:

```
var rules: [Rule] {  
  [  
    #Rule(Self.didTriggerTapTextEvent) {  
      $0.donations.donatedWithin(.day).count > 2  
    }  
  ]  
}
```

Eventos con valores asociados

```
import TipKit

struct Info: Codable, Sendable {
    let id: Int
    let data: String
}

struct MyTip: Tip {
    ...
    static let didTriggerTapTextEvent = Event(id: "event")
}
```

Eventos con valores asociados

```
import TipKit

struct MyTip: Tip {
    ...
    static let didTriggerTapTextEvent = Event<Info>(id: "event")
}
```

Eventos con valores asociados

```
import TipKit

struct MyTip: Tip {
    ...
    static let didTriggerTapTextEvent = Event<Info>(id: "event")

    var rules: [Rule] {
        [
            #Rule(Self.didTriggerTapTextEvent) {
                $0.donations.filter({ $0.data == "tap" }).count > 2
            }
        ]
    }
}
```


Eventos con valores asociados

Ejemplo de cómo usarlo:

```
MyTip.didTapEvent.donate(.init(id: 1, data: "tap"))
```

Combinar reglas

```
import TipKit

struct MyTip: Tip {
    ...
    var rules: [Rule] {
        [
            #Rule(Self.$isCounterTapped) { $0 == true },
            #Rule(Self.didTriggerTapTextEvent) { $0.donations.count > 2 }
        ]
    }
}
```

Estado de un Tip

- 1 Disponible para mostrar:
`available`
- 2 No es válido:
`invalidated(Tips.InvalidReason)`
- 3 No es elegible para mostrar:
`pending`

Obtener estado actual:

```
1 var status: Self.Status { get }
```

```
2 var statusUpdates: AsyncStream<Self.Status> { get }
```

Conocer si el tip está disponible para mostrarse:

```
1 var shouldDisplay: Bool { get }
```

```
2 var shouldDisplayUpdates:  
    AsyncMapSequence<AsyncStream<Self.Status>, Bool> { get }
```

Otras configuraciones

Almacenar datos de los tips

```
try? Tips.configure([  
    .datastoreLocation(.applicationDefault)  
])
```

Almacenar datos de los tips

```
try? Tips.configure([  
    .datastoreLocation(.applicationDefault)  
])
```

- 1 Directorio de soporte de la app:
applicationDefault
- 2 Contenedor de group de apps:
groupContainer(identifier: String)
- 3 Directorio personalizado:
url(_ url: URL)

Frecuencia de visualización general

```
try? Tips.configure([  
    .displayFrequency(.immediate)  
])
```

Frecuencia de visualización general

```
try? Tips.configure([  
    .displayFrequency(.immediate)  
])
```

Opciones:

- 1 **immediate**
- 2 **hourly**
- 3 **daily**
- 4 **weekly**
- 5 **monthly**

Frecuencia de visualización por Tip

```
import TipKit

struct MyTip: Tip {
    var options: [any TipOption] {
        IgnoresDisplayFrequency(true)
    }
}
```

Frecuencia de visualización por Tip

```
import TipKit

struct MyTip: Tip {
    var options: [any TipOption] {
        MaxDisplayCount(1)
    }
}
```

Testing

1 Mostrar todos los tips.
`Tips.showAllTipsForTesting()`

1 Mostrar todos los tips.

```
Tips.showAllTipsForTesting()
```

2 Mostrar algunos tips.

```
Tips.showTipsForTesting(_ tips: [any Tip.Type])
```

1 Mostrar todos los tips.

```
Tips.showAllTipsForTesting()
```

2 Mostrar algunos tips.

```
Tips.showTipsForTesting(_ tips: [any Tip.Type])
```

3 Ocultar todos los Tips.

```
Tips.hideAllTipsForTesting()
```


1 Mostrar todos los tips.

```
Tips.showAllTipsForTesting()
```

2 Mostrar algunos tips.

```
Tips.showTipsForTesting(_ tips: [any Tip.Type])
```

3 Ocultar todos los Tips.

```
Tips.hideAllTipsForTesting()
```

4 Ocultar algunos los Tips.

```
Tips.hideTipsForTesting(_ tips: [any Tip.Type])
```


- 1 Mostrar todos los tips.
`Tips.showAllTipsForTesting()`
- 2 Mostrar algunos tips.
`Tips.showTipsForTesting(_ tips: [any Tip.Type])`
- 3 Ocultar todos los Tips.
`Tips.hideAllTipsForTesting()`
- 4 Ocultar algunos los Tips.
`Tips.hideTipsForTesting(_ tips: [any Tip.Type])`
- 5 Eliminar todos los datos relacionados con TipKit y restablece el estado de todos los Tips.
`Tips.resetDatastore()`


- 1 Mostrar todos los tips.
`com.apple.TipKit.ShowAllTips 1`
- 2 Mostrar algunos tips.
`com.apple.TipKit.ShowTips tipID, otherTipID`
- 3 Ocultar todos los Tips.
`com.apple.TipKit.HideAllTips 1`
- 4 Ocultar algunos los Tips.
`com.apple.TipKit.HideTips tipID, otherTipID`
- 5 Eliminar todos los datos relacionados con TipKit y restablece el estado de todos los Tips.
`com.apple.TipKit.ResetDatastore 1`


Base de datos de TipKit

Archivos de TipKit

 tips-defaults.plist

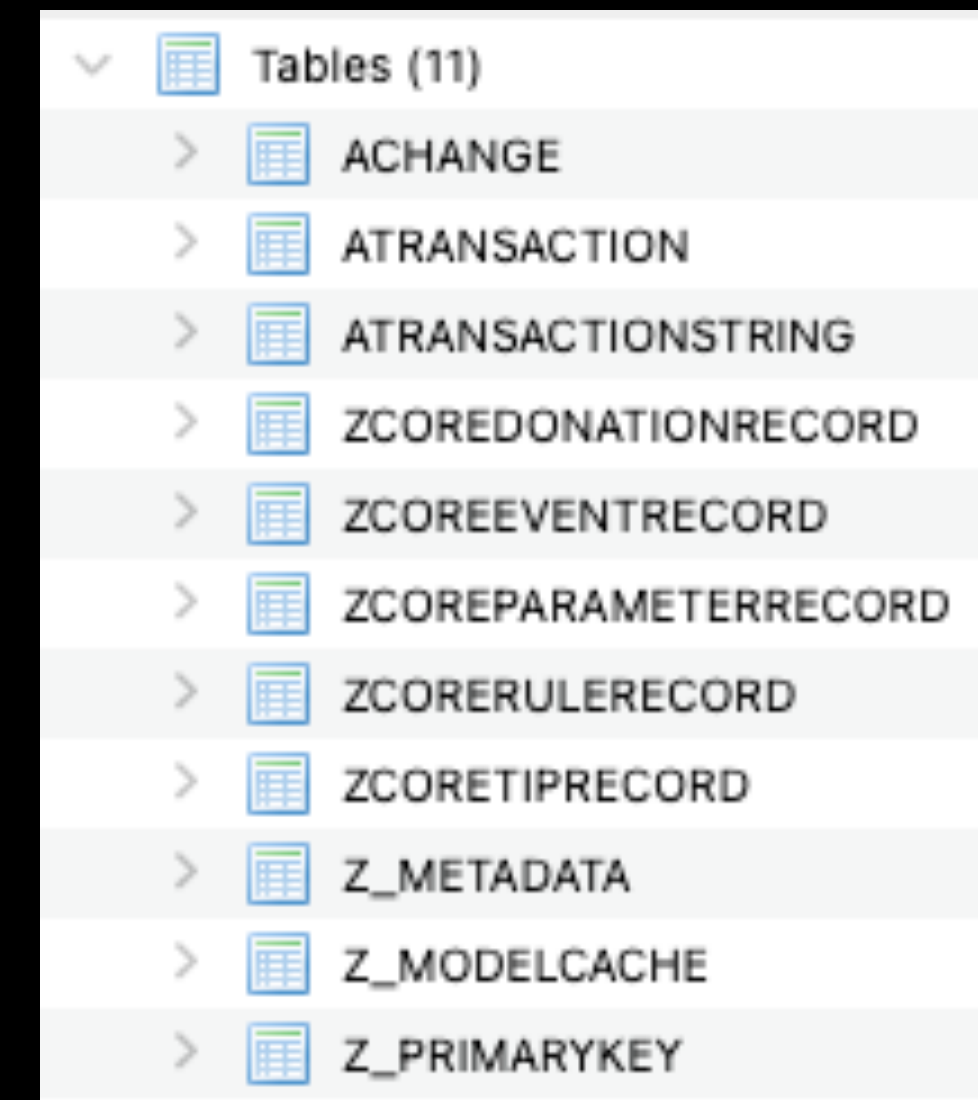
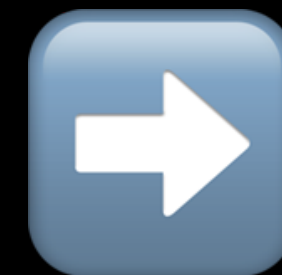
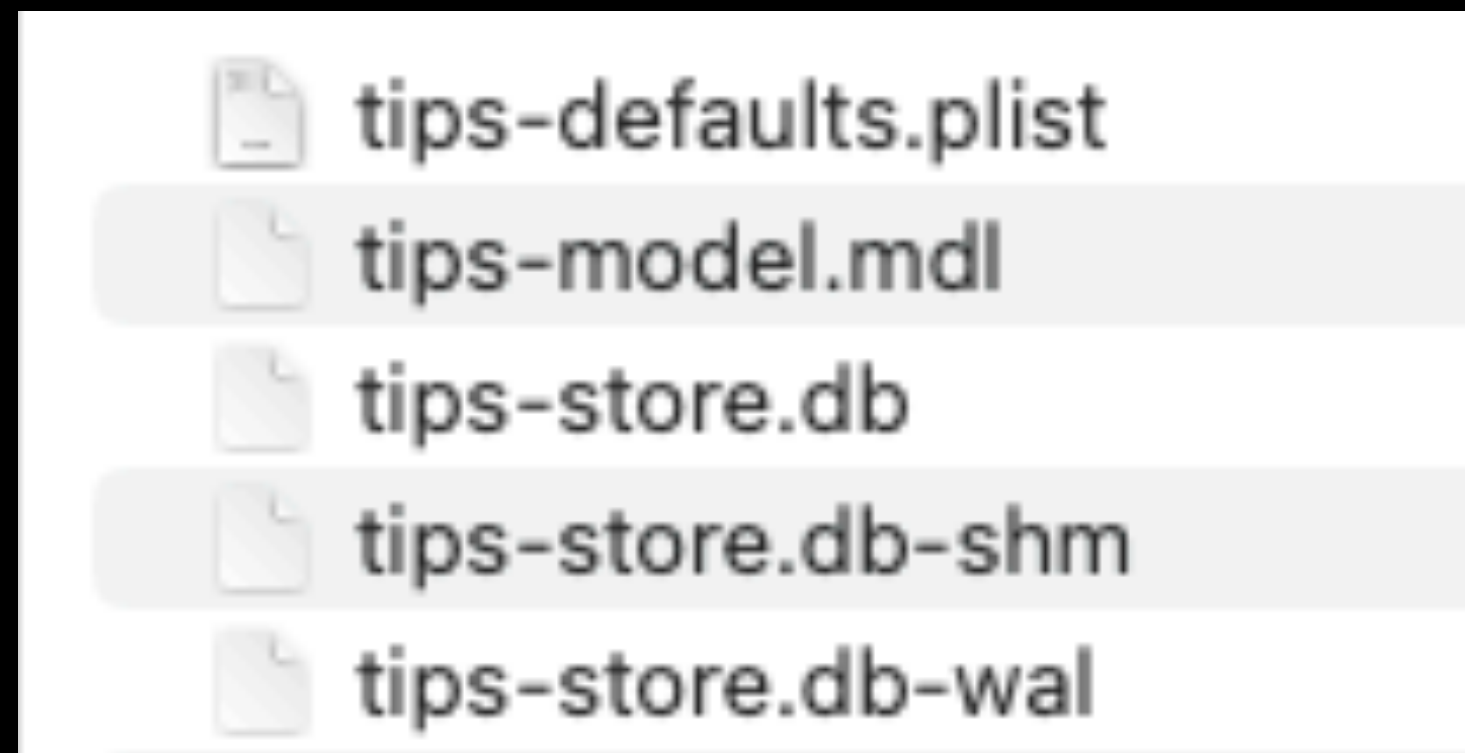
 tips-model.mdl

 tips-store.db

 tips-store.db-shm

 tips-store.db-wal

Archivos de TipKit



TipKit + UIKit

- 1 **Mostrar un inline tip:**
`TipUIView`
- 2 **Mostrar un popover tip:**
`TipUIPopoverViewController`
- 3 **Mostrar un tip en un collection view:**
`TipUICollectionViewController`

TipKit + AppKit

- 1 Mostrar un inline tip:
TipNSView
- 2 Mostrar un popover tip:
TipNSPopover

Recomendaciones

Se recomienda usar tips que sean:

- ✓ Accionables.
- ✓ Fácil de recordar.
- ✓ Instructivos.

NO se recomienda usar tips para:

- ✗ Promocionar mensajes.
- ✗ Mostrar mensajes de error.
- ✗ Muy complicados en su formato.

Más información

1 Sugerir funcionalidades a los usuarios con TipKit

<https://asynclearn.com/blog/sugerir-funcionalidades-tipkit/>

2 Definir reglas para mostrar tips en TipKit

<https://asynclearn.com/blog/reglas-tipkit/>

3 Documentación oficial de Apple

<https://developer.apple.com/documentation/tipkit>

5 Guía de interfaz de Apple

<https://developer.apple.com/design/human-interface-guidelines/offering-help>