

NSBuilders

Creando Lotería en SwiftUI

Lorenzo Brown - 21 de marzo 2024

La Misión

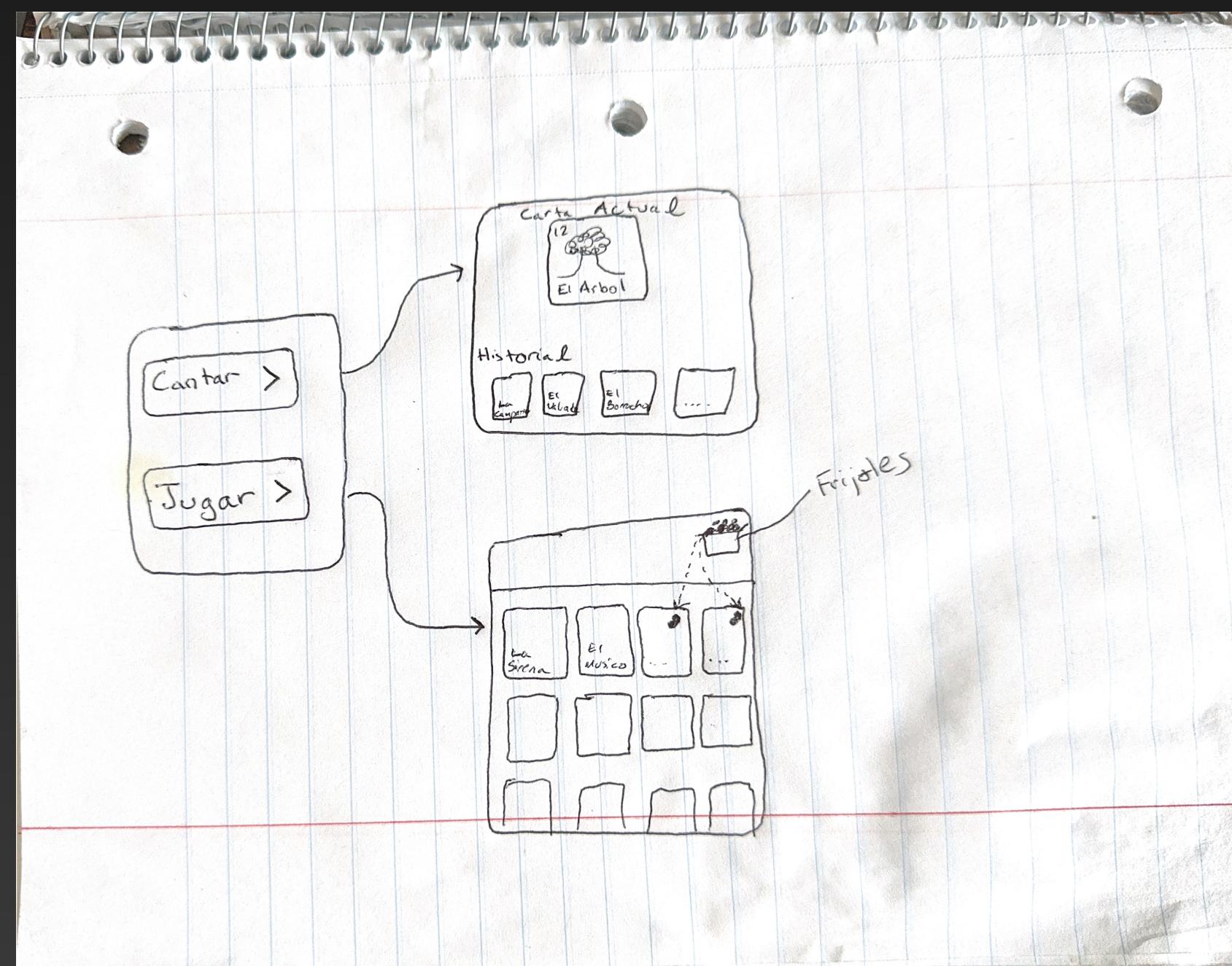
Crear una app de Lotería

- Versión Básica
 - Tablero de cartas
 - Marcadores para seleccionar cada carta (
 - Una vista para cantar las cartas

Paso 1

Un dibujito...

- Dibuja un diseño simple de lo que vas a crear.
- No tiene que ser bonito pero ayuda a ver los componentes que crearás



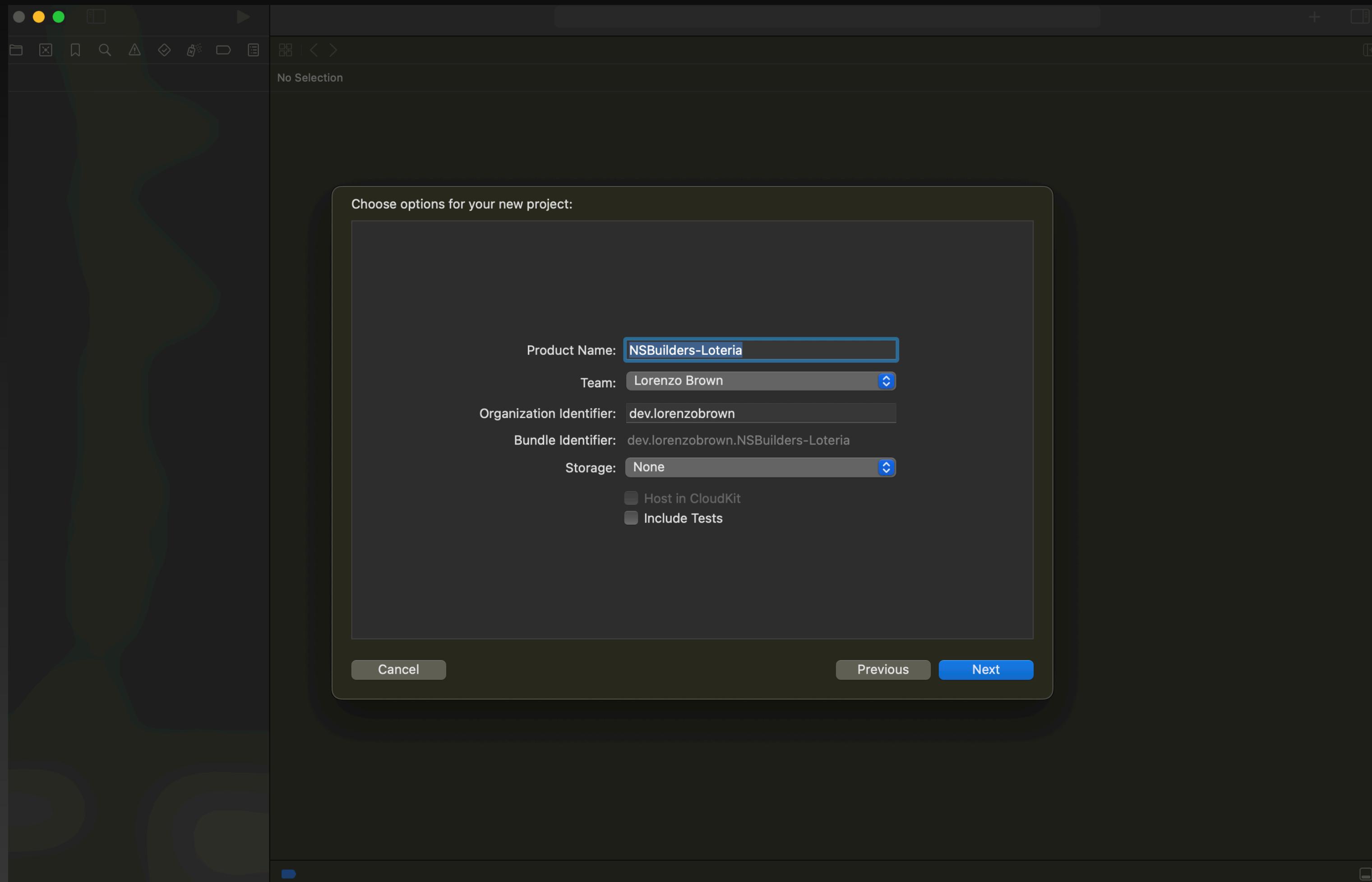
Paso 2

El Plan

- Haz un plan de los objetivos que quieres lograr
 - Mi Plan
 - Checkpoint 1 - Crear una app vacía en Xcode
 - Checkpoint 2 - Crear la vista de una Carta (CellView)
 - Checkpoint 3 - Crear un tablero de cartas (GridView)
 - Checkpoint 4 - Crear un Modelo y datos dinámicos
 - Checkpoint 5 - Crear una cabecera (HeaderView)
 - Checkpoint 6 - Arrastrar y Soltar los frijoles al tablero
 - Checkpoint 7 - Crear el modo de cantante (ChanterView)
 - Checkpoint 8 - Agregar un poco de estilo y efectos especiales

Checkpoint 1

Una app vacía en Xcode

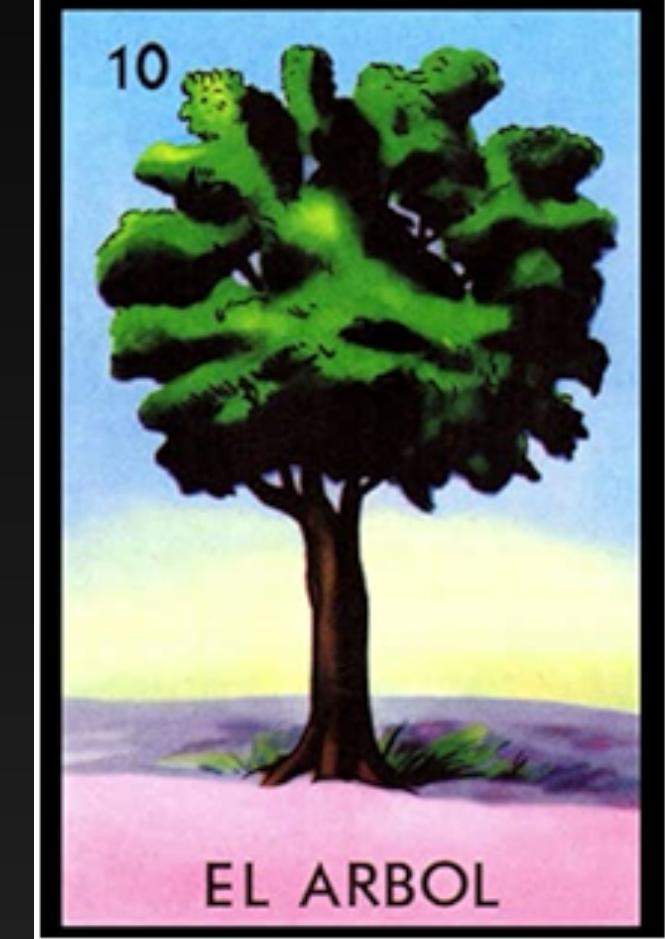


<https://github.com/lrnzbr/NSBuilders-Loteria/commit/81f44396d80d865ffe7cfa897cb69c3559ae9659>

Checkpoint 2

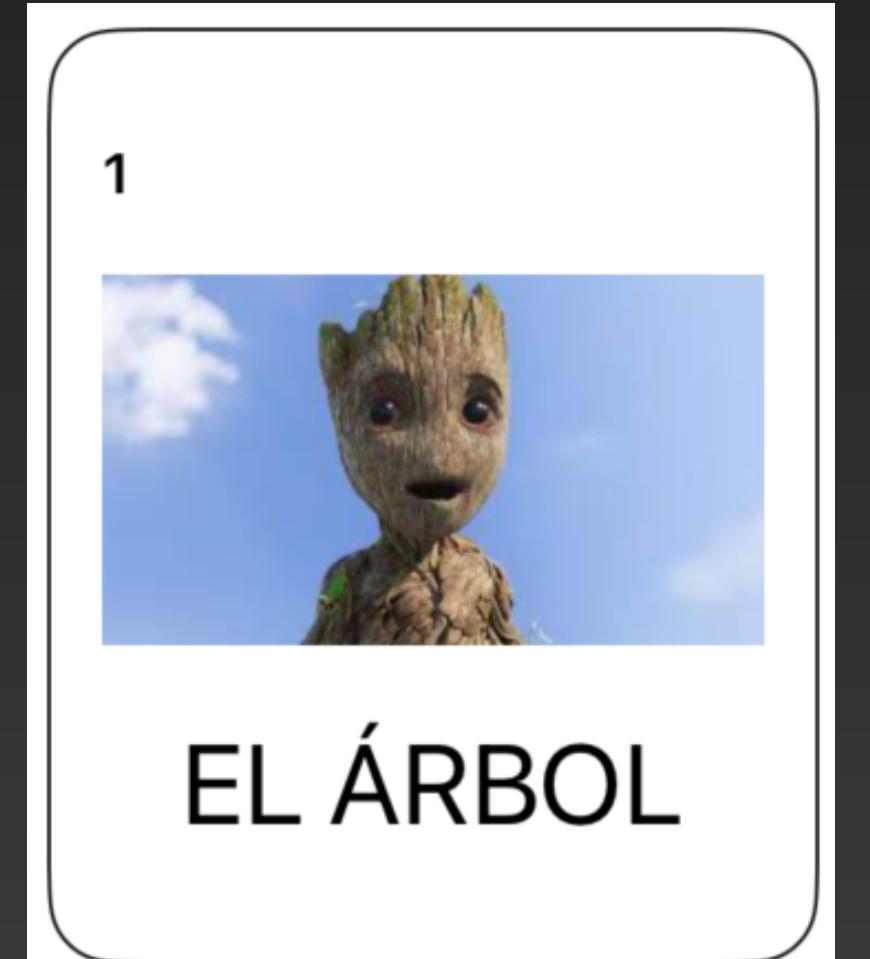
CellView.swift

Expectativa



```
7  
8 import SwiftUI  
9  
10 struct CellView: View {  
11     var body: some View {  
12         ZStack {  
13             VStack {  
14                 HStack {  
15                     Text("1")  
16                         .font(.headline)  
17                         .foregroundColor(.black)  
18                     Spacer()  
19                 }  
20                 VStack {  
21                     Image("arbol")  
22                         .resizable()  
23                         .clipped()  
24                         .aspectRatio(contentMode: .fit)  
25                         .frame(maxHeight: 130)  
26                 }.clipped()  
27                 Text("EL ÁRBOL")  
28                     .lineLimit(2)  
29                     .font(.largeTitle)  
30                     .minimumScaleFactor(0.2)  
31                     .multilineTextAlignment(.center)  
32             }  
33         }  
34         .frame(maxWidth: 200, maxHeight: 250)  
35         .padding()  
36         .background(.white)  
37         .cornerRadius(20)  
38         .overlay(  
39             RoundedRectangle(cornerRadius: 20)  
40                 .stroke(.black, lineWidth: 1)  
41         )  
42     }  
43 }  
44  
45 #Preview {  
46     CellView()  
47 }
```

Realidad



<https://github.com/lrnzbr/NSBuilders-Loteria/commit/2dd0338891d9090d35bba7bb2fb46fb9ad2f1b3>

Checkpoint 3

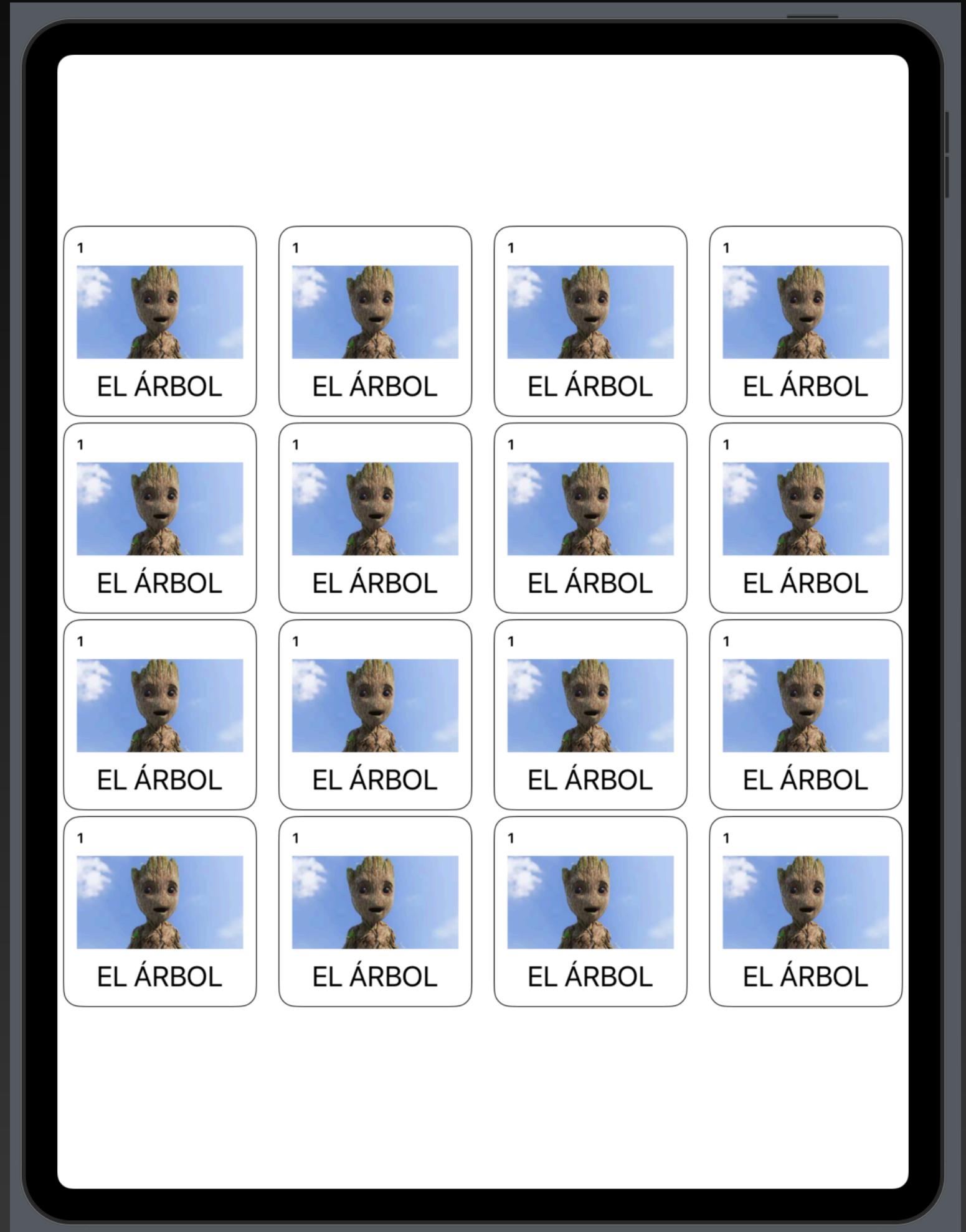
GridView.swift

```
struct GridView: View {

    let columns: [GridItem] = [
        GridItem(.flexible(), spacing: 12),
        GridItem(.flexible(), spacing: 12),
        GridItem(.flexible(), spacing: 12),
        GridItem(.flexible(), spacing: 12)
    ]

    var body: some View {
        GeometryReader { geo in
            ZStack {
                LazyVGrid(columns: columns) {
                    ForEach(0...15, id: \.self) { number in
                        CellView()
                            .frame(maxWidth: geo.size.width / 4, maxHeight:
                                geo.size.height / 4)
                    }
                }.padding(.bottom)
                    .frame(maxWidth: .infinity, maxHeight: .infinity)
            }
        }
    }

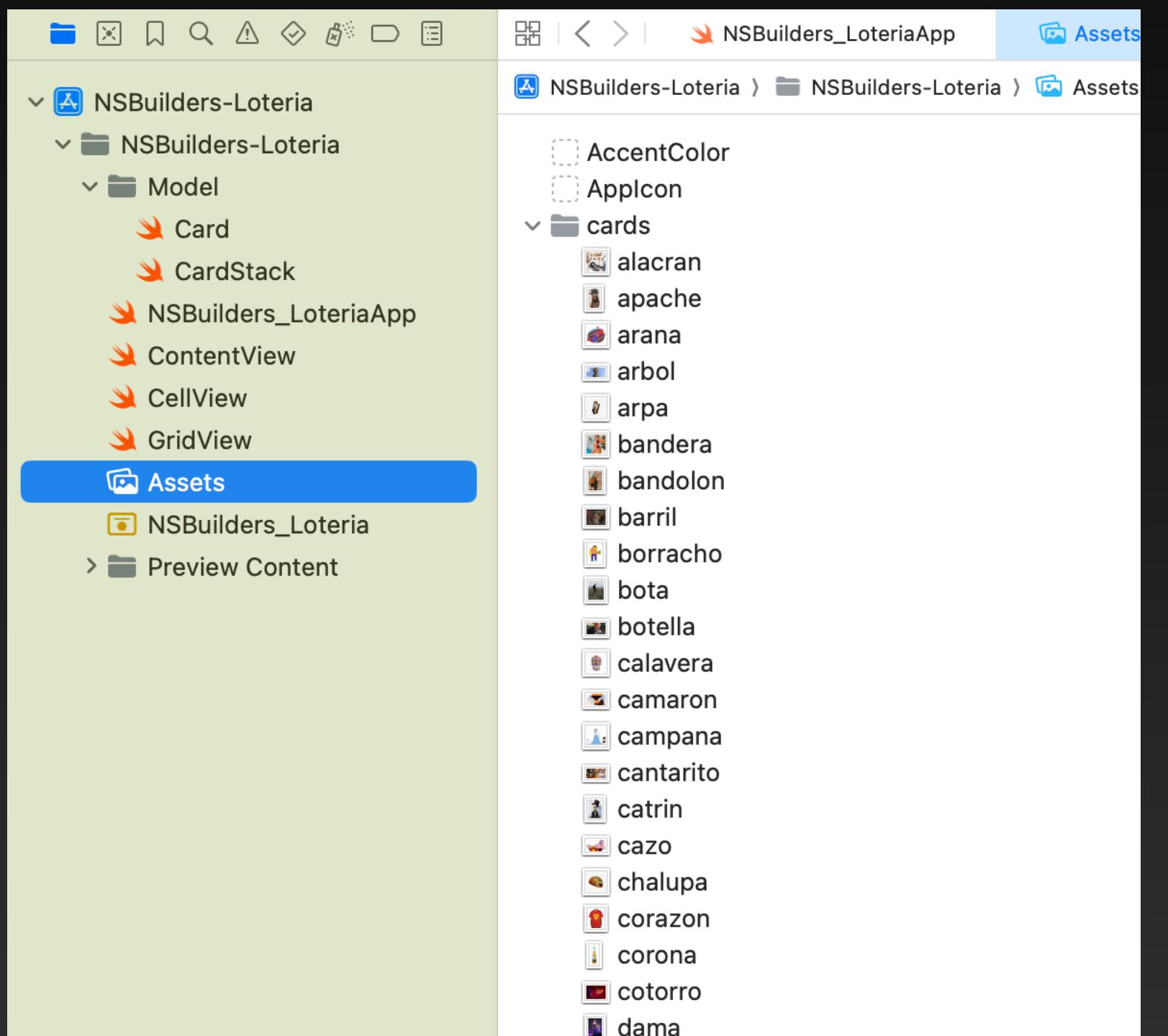
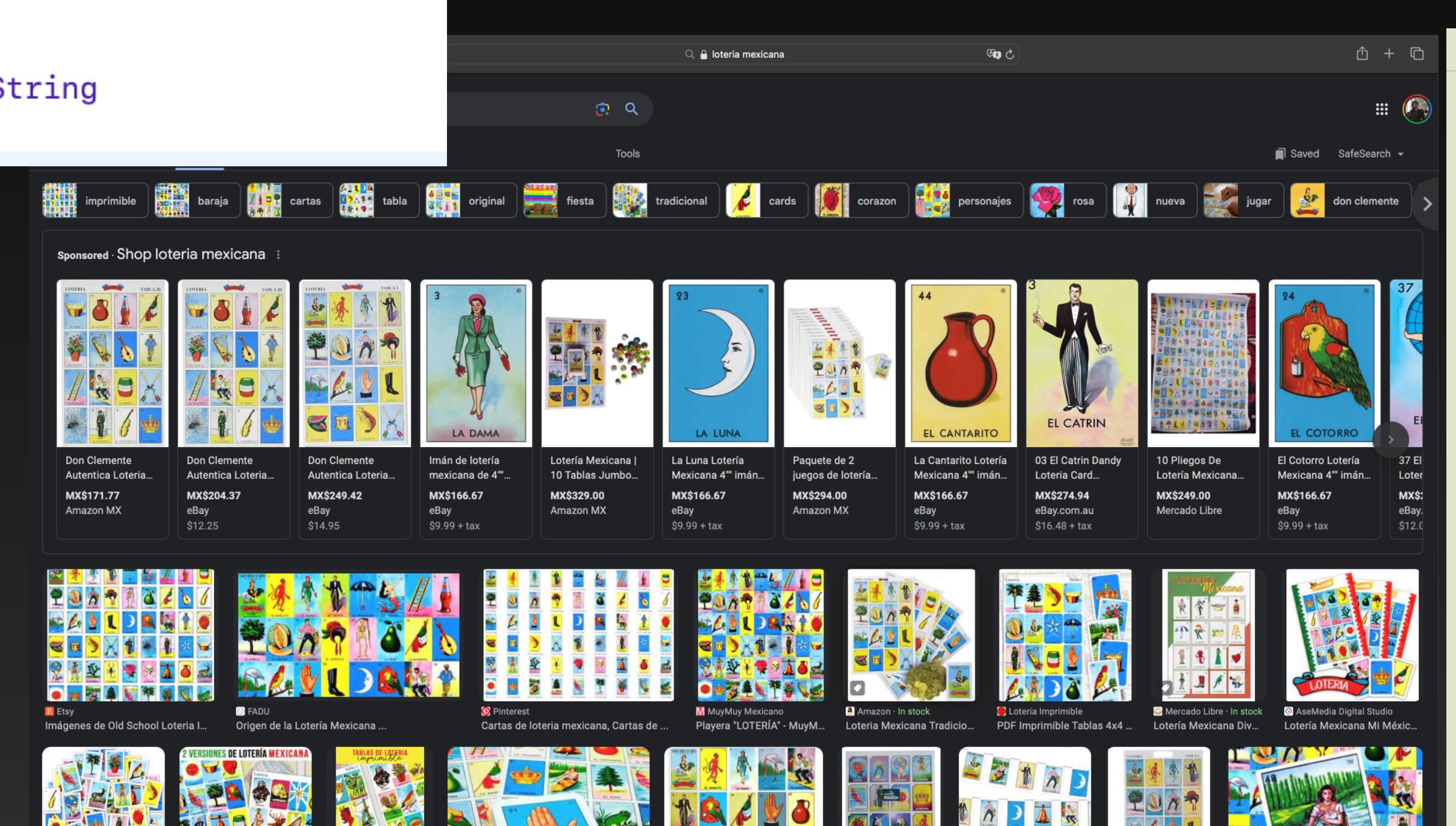
    #Preview {
        GridView()
    }
}
```



Checkpoint 4

Datos Dinámicos

```
8 struct Card : Identifiable, Hashable {  
9     var id: Int  
10    var title: String  
11    var imageString: String  
12 }
```

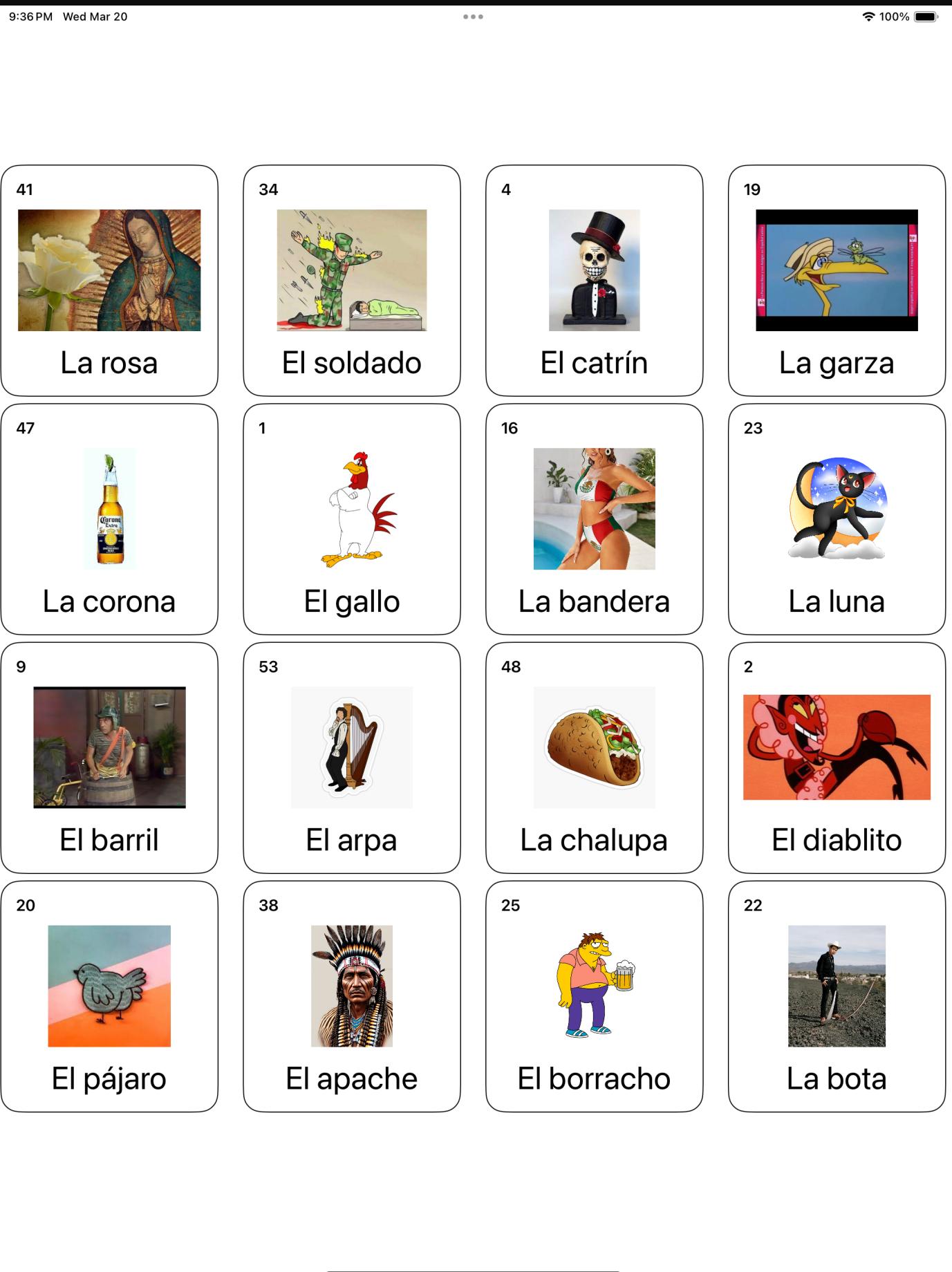


Checkpoint 4

Datos Dinámicos

```
1 //  
2 // CardStack.swift  
3 // NSBuilders-Loteria  
4 //  
5 // Created by Lorenzo Brown on 3/20/24.  
6 //  
7  
8 let cardStack : [Card] = [ card1, card2, card3, card4, card5, card6, card7, card8, card9, card10, card11, card12, card13, card14,  
    card15, card16, card17, card18, card19, card20, card21, card22, card23, card24, card25, card26, card27, card28, card29,  
    card30, card31, card32, card33, card34, card35, card36, card37, card38, card39, card40, card41, card42, card43, card44, card45,  
    card46, card47, card48, card49, card50, card51, card52, card53, card54]  
9  
10 let card1 = Card(id: 1, title: "El gallo", imageString: "gallo")  
11 let card2 = Card(id: 2, title: "El diablito", imageString: "diablito")  
12 let card3 = Card(id: 3, title: "La dama", imageString: "dama")  
13 let card4 = Card(id: 4, title: "El catrín", imageString: "catrin")  
14 let card5 = Card(id: 5, title: "El paraguas", imageString: "paraguas")  
15 let card6 = Card(id: 6, title: "La sirena", imageString: "sirena")  
16 let card7 = Card(id: 7, title: "La escalera", imageString: "escalera")  
17 let card8 = Card(id: 8, title: "La botella", imageString: "botella")  
18 let card9 = Card(id: 9, title: "El barril", imageString: "barril")  
19 let card10 = Card(id: 10, title: "El árbol", imageString: "arbol")  
20  
21 let card11 = Card(id: 11, title: "El melón", imageString: "melon")  
22 let card12 = Card(id: 12, title: "El valiente", imageString: "valiente")  
23 let card13 = Card(id: 13, title: "El gorrito", imageString: "gorrito")  
24 let card14 = Card(id: 14, title: "La muerte", imageString: "muerte")  
25 let card15 = Card(id: 15, title: "La pera", imageString: "pera")  
26 let card16 = Card(id: 16, title: "La bandera", imageString: "bandera")  
27 let card17 = Card(id: 17, title: "El bandolón", imageString: "bandolon")  
28 let card18 = Card(id: 18, title: "El violoncello", imageString: "violoncello")  
29 let card19 = Card(id: 19, title: "La garza", imageString: "garza")  
30 let card20 = Card(id: 20, title: "El pájaro", imageString: "pajaro")|
```

```
struct GridView: View {  
  
    let columns: [GridItem] = [  
        GridItem(.flexible(), spacing: 12),  
        GridItem(.flexible(), spacing: 12),  
        GridItem(.flexible(), spacing: 12),  
        GridItem(.flexible(), spacing: 12)  
    ]  
  
    var body: some View {  
        GeometryReader { geo in  
            ZStack {  
                LazyVGrid(columns: columns) {  
                    ForEach(0...15, id: \.self) { number in  
                        CellView(card: cardStack.randomElement()!)  
                            .frame(maxWidth: geo.size.width / 4, maxHeight:  
                                geo.size.height / 4)  
                    }  
                }.padding(.bottom)  
                .frame(maxWidth: .infinity, maxHeight: .infinity)  
            }  
        }  
    }  
}
```



Checkpoint 5

HeaderView.swift

[https://github.com/lrnzbr/NSBuilders-Loteria/commit/
d83502c4eb8573f996b6837d12d6cde9fb22b359](https://github.com/lrnzbr/NSBuilders-Loteria/commit/d83502c4eb8573f996b6837d12d6cde9fb22b359)

```
func createPlayingCard(sampleCards: [Card], numberOfCards:Int = 16)->[Card] {
    guard sampleCards.count >= numberOfCards else {
        return []
    }
    var output:Set = [sampleCards.randomElement()!]

    while output.count < numberOfCards {
        output.insert(sampleCards.randomElement()!)
    }

    let outputArray = Array(output)
    return outputArray
}
```

```
8 import SwiftUI
9
10 struct HeaderView: View {
11     @Binding var playingCards:[Card]
12
13     func resetCards() {
14         playingCards = createPlayingCard(sampleCards: cardStack)
15     }
16
17     var body: some View {
18         VStack {
19             HStack {
20                 Button("Crear Nueva Carta"){
21                     resetCards()
22                 }.padding()
23                     .foregroundColor(.white)
24                     .background(Color(red: 0, green: 0, blue: 0.5))
25                     .clipShape(Capsule())
26
27                 Spacer()
28             }.padding()
29         }
30     }
31
32 #Preview {
33     HeaderView(playingCards: .constant(Array(cardStack[0...15])))
34 }
```

Checkpoint 5

HeaderView.swift

[https://github.com/lrnzbr/NSBuilders-Loteria/commit/
d83502c4eb8573f996b6837d12d6cde9fb22b359](https://github.com/lrnzbr/NSBuilders-Loteria/commit/d83502c4eb8573f996b6837d12d6cde9fb22b359)

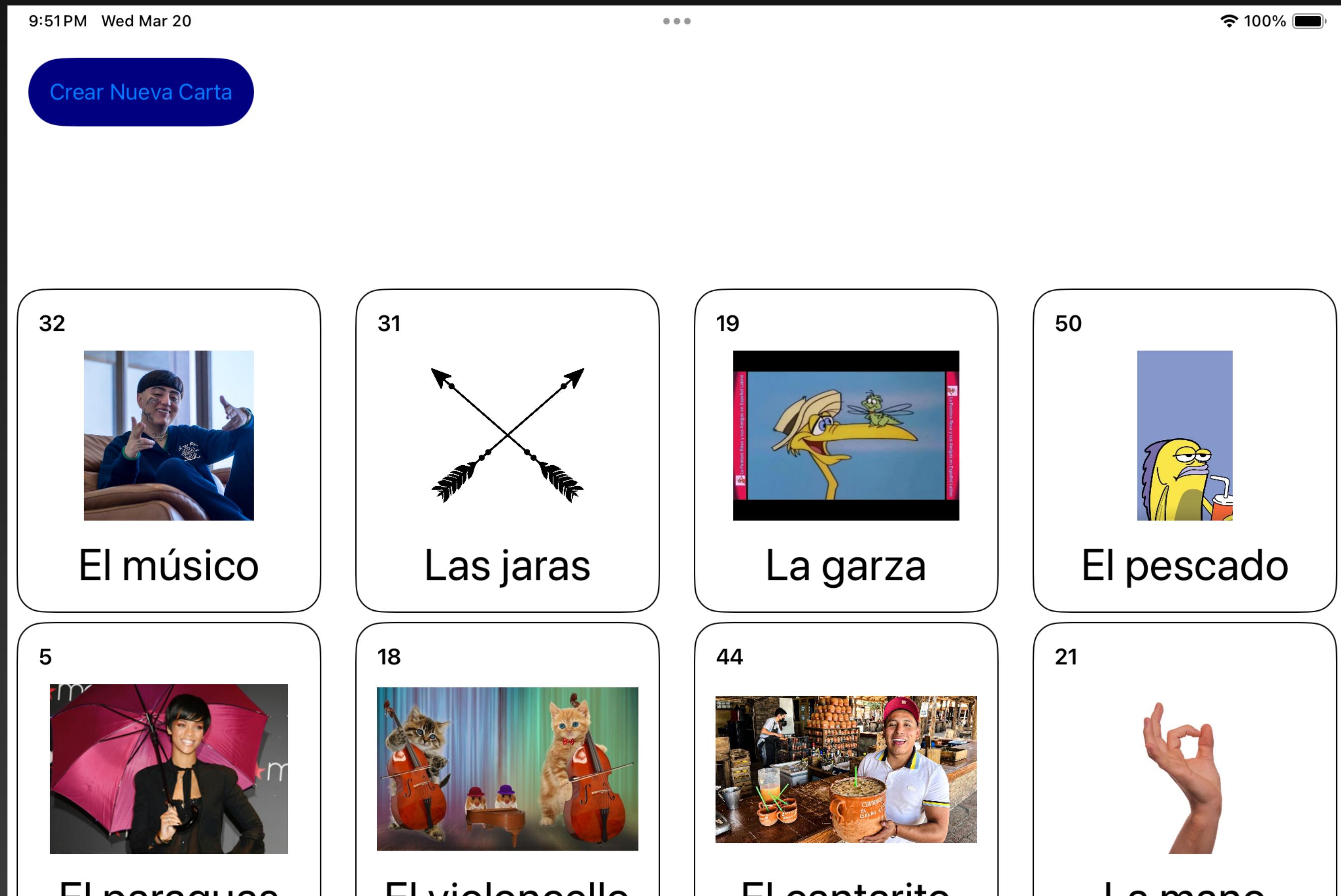
```
9
10 struct PlayView: View {
11     @State var playingCards:[Card] = createPlayingCard(sampleCards: cardStack)
12
13     var body: some View {
14         VStack {
15             HeaderView(playingCards: $playingCards)
16             GridView(playingCards: $playingCards)
17         }
18     }
19 }
```

```
/  
8 import SwiftUI  
9  
10 struct GridView: View {  
11     @Binding var playingCards:[Card]  
12
13     let columns: [GridItem] = [  
14         GridItem(.flexible(), spacing: 12),  
15         GridItem(.flexible(), spacing: 12),  
16         GridItem(.flexible(), spacing: 12),  
17         GridItem(.flexible(), spacing: 12)  
18     ]  
19
20     var body: some View {
21         GeometryReader { geo in
22             ZStack {
23                 LazyVGrid(columns: columns)
24             {
25                 ForEach(0...15, id: \.self) { number in
26                     CellView(card: playingCards[number])
27                         .frame(maxWidth: geo.size.width / 4, maxHeight: geo.size.height / 4)
28                 }
29             }
29 }
```

Checkpoint 5

HeaderView.Swift

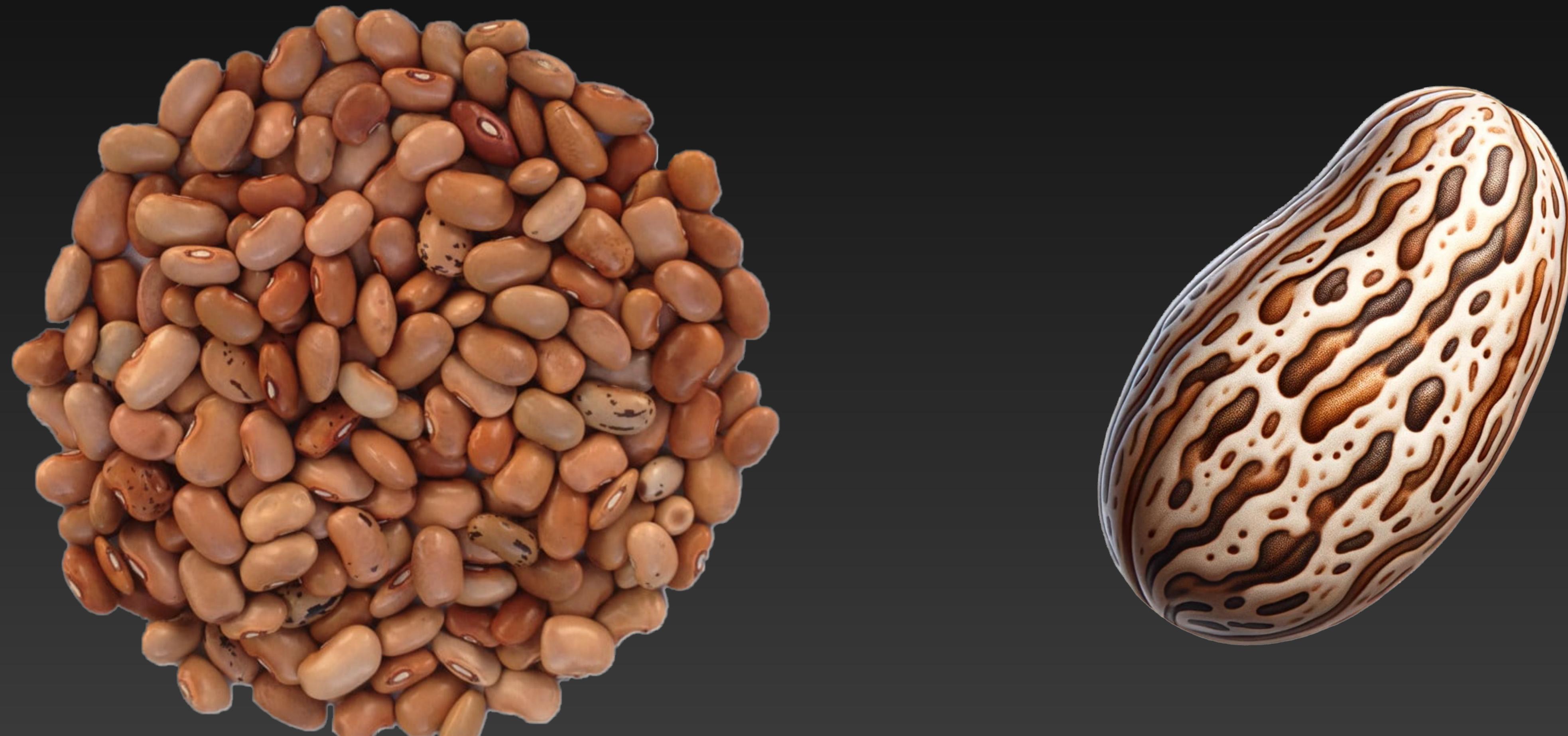
[https://github.com/Irnzbr/NSBuilders-Loteria/commit/
d83502c4eb8573f996b6837d12d6cde9fb22b359](https://github.com/Irnzbr/NSBuilders-Loteria/commit/d83502c4eb8573f996b6837d12d6cde9fb22b359)



Checkpoint 6

“Drag-n-Drop”

<https://github.com/lrnzbr/NSBuilders-Loteria/commit/d9b446fcc4e83dfdc81950a9b0d4e69e0714fbfe>



Checkpoint 6

“Drag-n-Drop”

<https://github.com/lrnzbr/NSBuilders-Loteria/commit/d9b446fcc4e83dfdc81950a9b0d4e69e0714fbfe>

Protocol

Transferable

A protocol that describes how a type interacts with transport APIs such as drag and drop or copy and paste.

iOS 16.0+ iPadOS 16.0+ macOS 13.0+ Mac Catalyst 16.0+ tvOS 16.0+ watchOS 9.0+ visionOS 1.0+

```
protocol Transferable
```

Checkpoint 6

“Drag-n-Drop”

<https://github.com/lrnzbr/NSBuilders-Loteria/commit/d9b446fcc4e83dfdc81950a9b0d4e69e0714fbfe>

```
9
10 struct HeaderView: View {
11     @Binding var beansInBucket:[String]
12     @Binding var beansInSquare:[Int: [String]]
13     @Binding var playingCards:[Card]
14
15     func resetCards() {
16         beansInBucket = ["Bean1", "Bean2", "Bean3", "Bean4", "Bean 5", "Bean6", "Bean7", "Bean8", "Bean9", "Bean10", "Bean11", "Bean12",
17             "Bean13", "Bean14", "Bean15", "Bean16"]
18         beansInSquare.removeAll()
19
20         playingCards = createPlayingCard(sampleCards: cardStack)
21     }
22
23     var body: some View {
24         VStack {
25             HStack {
26                 Button("Crear Nueva Carta"){
27                     resetCards()
28                 }.padding()
29                     .foregroundColor(.white)
30                     .background(Color(red: 0, green: 0, blue: 0.5))
31                     .clipShape(Capsule())
32             Spacer()
33             Text("Arrastra tus frijoles →")
34                 .font(.caption2)
35                 .lineLimit(2)
36                 .multilineTextAlignment(.trailing)
37             ZStack {
38                 Image("beans")
39                     .resizable()
40                     .aspectRatio(contentMode: .fit)
41                     .frame(maxHeight: 80)
42            ForEach(beansInBucket, id: \.self){ bean in
43                 Image("bean")
44                     .resizable()
45                     .frame(width: 30, height: 30)
46                     .draggable(bean)
47             }
48         }
49         .dropDestination(for: String.self) { droppedBeans, location in
50             //add a bean back to the bucket
51             for bean in droppedBeans{
52                 beansInBucket.append(bean)
53                 //remove from square
54                 for square in beansInSquare.keys {
55                     beansInSquare[square]?.removeAll{$0 == bean}
56                 }
57             }
58         }
59     }
60 }
```

Checkpoint 6

“Drag-n-Drop”

<https://github.com/lrnzbr/NSBuilders-Loteria/commit/d9b446fcc4e83dfdc81950a9b0d4e69e0714fbfe>

```
10 struct GridView: View {
11     @Binding var beansInBucket:[String]
12     @Binding var beansInSquare:[Int: [String]]
13     @Binding var playingCards:[Card]
14
15     let columns: [GridItem] = [
16         GridItem(.flexible(), spacing: 12),
17         GridItem(.flexible(), spacing: 12),
18         GridItem(.flexible(), spacing: 12),
19         GridItem(.flexible(), spacing: 12)
20     ]
21
22     var body: some View {
23         GeometryReader { geo in
24             ZStack {
25                 LazyVGrid(columns: columns)
26                 {
27                     ForEach(0...15, id: \.self) { number in
28                         CellView(beans: beansInSquare[number] ?? [], card: playingCards[number])
29                             .frame(maxWidth: geo.size.width / 4, maxHeight: geo.size.height / 4)
30                             .dropDestination(for: String.self) { droppedBeans, location in
31                                 // remove from bucket
32                                 for bean in droppedBeans {
33                                     beansInBucket.removeAll {$0 == bean}
34                                     //remove drag from other squares
35                                     for square in beansInSquare.keys {
36                                         beansInSquare[square]?.removeAll{$0 == bean}
37                                     }
38                                 }
39                                 //add to square
40                                 var currentBeans = beansInSquare[number]
41                                 if let firstBean = droppedBeans.first {
42                                     if currentBeans != nil {
43                                         currentBeans?.append(firstBean)
44                                     } else {
45                                         currentBeans = [firstBean]
46                                     }
47                                     beansInSquare[number] = currentBeans
48                                 }
49                                 return true
50                             }
51             }
52         }
53     }
54 }
```

Checkpoint 6

“Drag-n-Drop”

<https://github.com/lrnzbr/NSBuilders-Loteria/commit/d9b446fcc4e83dfdc81950a9b0d4e69e0714fbfe>

```
10 struct PlayView: View {
11     @State private var beansInBucket:[String] = ["Bean1", "Bean2", "Bean3", "Bean4", "Bean 5", "Bean6", "Bean7", "Bean8", "Bean9",
12     "Bean10", "Bean11", "Bean12", "Bean13", "Bean14", "Bean15", "Bean16"]
13     @State private var beansInSquare: [Int: [String]] = [:]
14     @State var playingCards:[Card] = createPlayingCard(sampleCards: cardStack)
15
16     var body: some View {
17         VStack {
18             HeaderView(beansInBucket: $beansInBucket, beansInSquare: $beansInSquare, playingCards: $playingCards)
19             GridView(beansInBucket: $beansInBucket, beansInSquare: $beansInSquare, playingCards: $playingCards)
20         }
21     }
}
```

Checkpoint 6

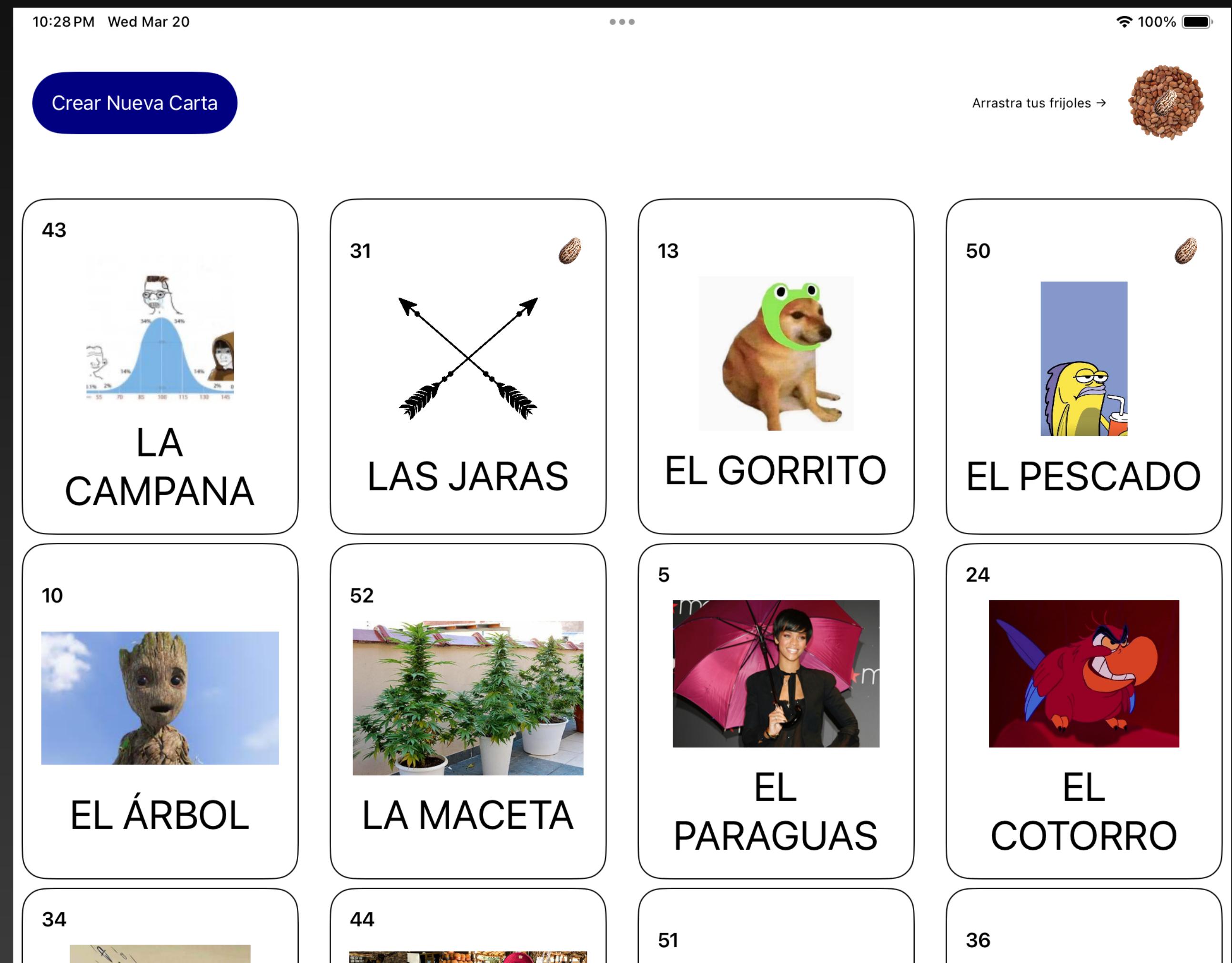
“Drag-n-Drop”

<https://github.com/lrnzbr/NSBuilders-Loteria/commit/d9b446fcc4e83dfdc81950a9b0d4e69e0714fbfe>

Checkpoint 6

“Drag-n-Drop”

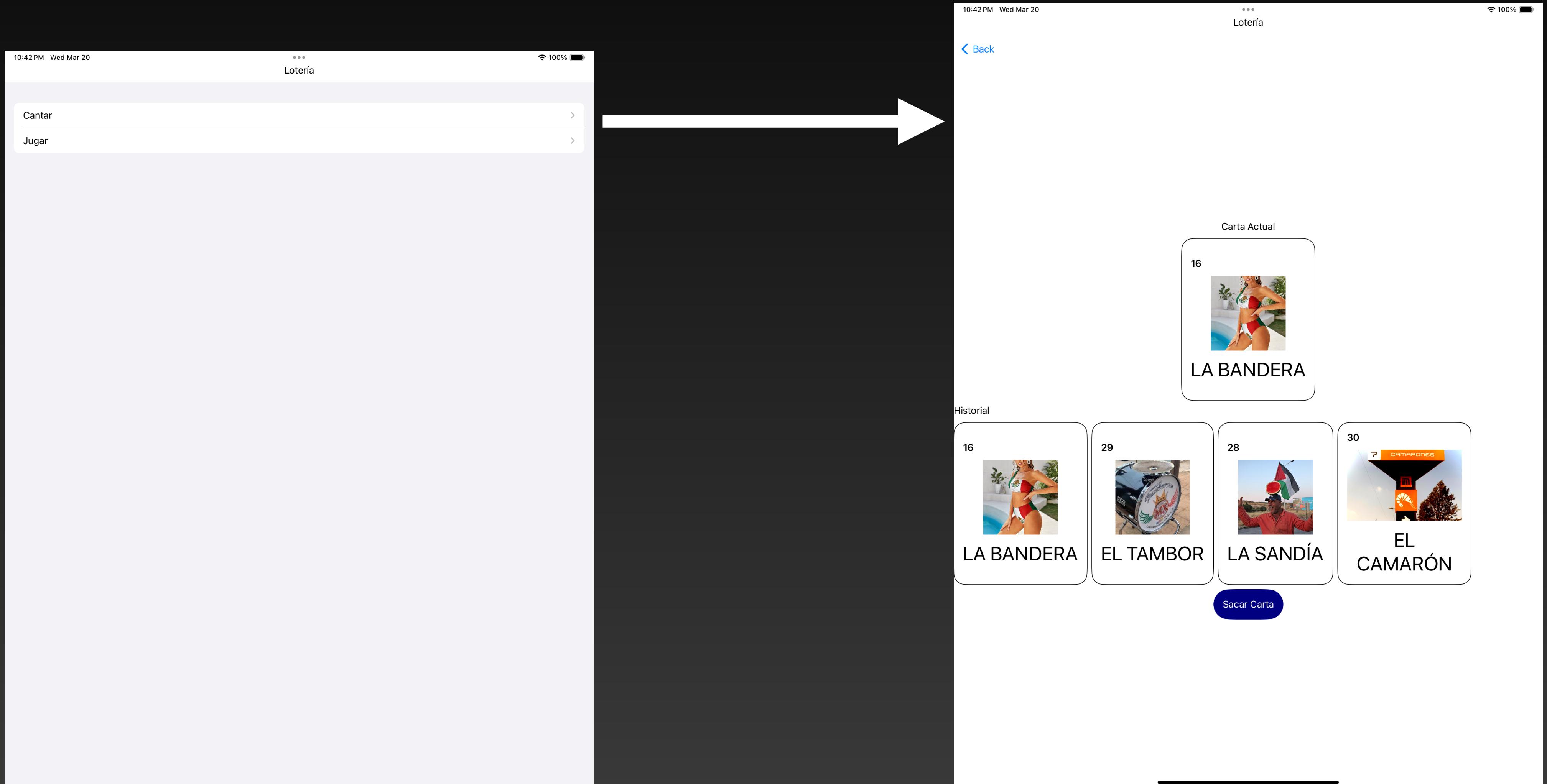
<https://github.com/lrnzbr/NSBuilders-Loteria/commit/d9b446fcc4e83dfdc81950a9b0d4e69e0714fbfe>



Checkpoint 7

ChanterView.Swift

<https://github.com/lrnzbr/NSBuilders-Loteria/commit/013520e07e437955b2bf59e9d11ef43f04052e94>



Checkpoint 7

ChanterView.swift

<https://github.com/lrnzbr/NSBuilders-Loteria/commit/013520e07e437955b2bf59e9d11ef43f04052e94>

```
import SwiftUI

struct ChanterView: View {
    let fullDeck: Set<Card> = Set(cardStack)
    @State var usedDeck = [Card]()

    func drawNewCard(){
        let newCard = fullDeck.randomElement()
        if let newCard = newCard {
            if usedDeck.contains(newCard){
                return drawNewCard()
            } else {
                usedDeck.append(newCard)
            }
        }
    }

    var body: some View {
        VStack {
            if let currentCard = usedDeck.last {
                Text("Carta Actual")
                CellView(beans: [], card: currentCard)
            }
            if usedDeck.count > 0 {
                HStack {
                    Text("Historial")
                    Spacer()
                }
            }
            ScrollView(.horizontal) {
                HStack {
                    ForEach(usedDeck.reversed()) { card in
                        CellView(beans: [], card: card)
                    }
                }
            }
        }
        Button("Sacar Carta"){
            drawNewCard()
        }.disabled(usedDeck.count >= fullDeck.count)
        .padding()
        .background(Color(red: 0, green: 0, blue: 0.5))
        .clipShape(Capsule())
        .foregroundColor(.white)
    }
}
```

Checkpoint 7

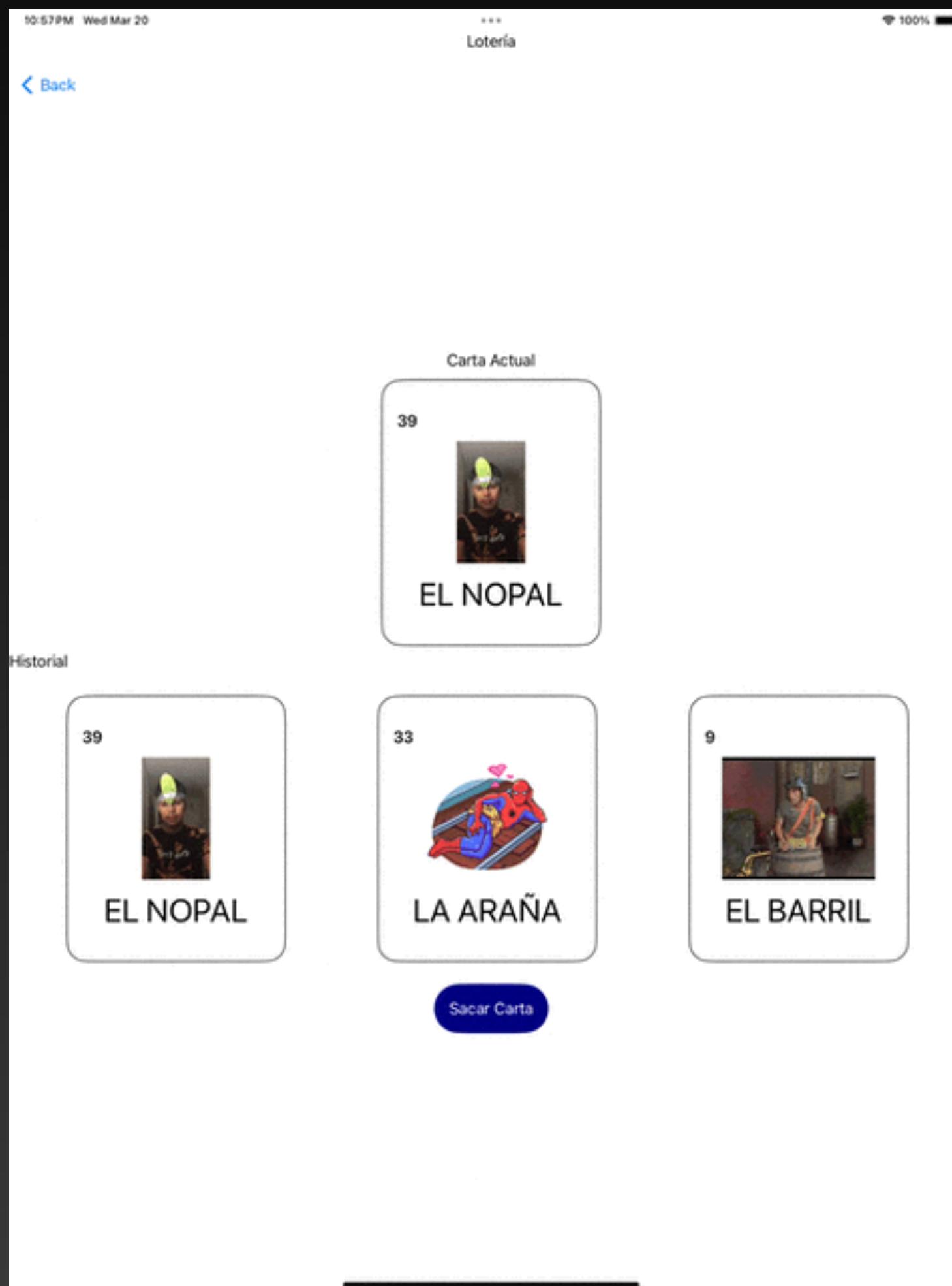
ChanterView.swift

[https://github.com/lrnzbr/
NSBuilders-Loteria/commit/
013520e07e437955b2bf59e9d11ef43
f04052e94](https://github.com/lrnzbr/NSBuilders-Loteria/commit/013520e07e437955b2bf59e9d11ef43f04052e94)

```
struct GameModeView: View {
    var body: some View {
        Text("Lotería")
        NavigationView{
            List {
                NavigationLink(destination: ChanterView()){
                    Text("Cantar")
                }
                NavigationLink(destination: PlayView()){
                    Text("Jugar")
                }
            }
            .navigationViewStyle(StackNavigationViewStyle())
        }
    }
}
```

Checkpoint 8

Un poco de estilo



<https://github.com/lrnzbr/NSBuilders-Loteria/commit/1823033264b29106d616c8e46b71ab9ee954d02e>

Checkpoint 8

Un poco de estilo

Sonido después de soltar el frijol

```
7
8 import SwiftUI
9 import AVFoundation
10
11 struct GridView: View {
12     @Binding var imageName: String
```

```
48     beansInSquare[hamper] = calicoBeans;
49
50     }  
51     AudioServicesPlaySystemSound(1022);
52     return true
53
54 }
```

<https://github.com/lrnzbr/NSBuilders-Loteria/commit/1823033264b29106d616c8e46b71ab9ee954d02e>

Checkpoint 8

Un poco de estilo

Cantar las cartas

```
8 import SwiftUI
9 import AVFoundation
10
11 var synthesizer = AVSpeechSynthesizer()
12
13 struct ChanterView: View {
14     let fullDeck: Set<Card> = Set(cardStack)
15     @State var usedDeck = [Card]()
16     @Environment(\.verticalSizeClass) var verticalSizeClass
17
18
19 func drawNewCard(){
20     let newCard = fullDeck.randomElement()
21     if let newCard = newCard {
22         if usedDeck.contains(newCard){
23             return drawNewCard()
24         } else {
25             usedDeck.append(newCard)
26             let utterance = AVSpeechUtterance(string: newCard.title)
27             utterance.voice = AVSpeechSynthesisVoice(language: "es-MX")
28             synthesizer.speak(utterance)
29         }
30     }
31 }
```

<https://github.com/lrnzbr/NSBuilders-Loteria/commit/1823033264b29106d616c8e46b71ab9ee954d02e>

Checkpoint 8

Un poco de estilo

Animar al desplazar las cartas (iOS 17+)

```
44 |     ScrollView(.horizontal) {
45 |         HStack {
46 |             ForEach(usedDeck.reversed()) { card in
47 |                 CellView(beans: [], card: card)
48 |                     .containerRelativeFrame(.horizontal, count: verticalSizeClass == .regular ? 3 : 5, spacing: 12)
49 |                     .scrollTransition { content, phase in
50 |                         content
51 |                             .opacity(phase.isIdentity ? 1.0 : 0.1)
52 |                             .scaleEffect(x: phase.isIdentity ? 1.0 : 0.3,
53 |                                         y: phase.isIdentity ? 1.0 : 0.3)
54 |                             .offset(y: phase.isIdentity ? 0 : 50)
55 |                     }
56 |             }
57 |             }.scrollTargetLayout()
58 |         }.contentMargins(16, for: .scrollContent)
59 |         .scrollTargetBehavior(.viewAligned)
```

<https://github.com/lrnzbr/NSBuilders-Loteria/commit/1823033264b29106d616c8e46b71ab9ee954d02e>

¿i y de aquí a dónde?!

- Métele diseño
 - ...o contrata a alguien que sabe de esas cosas
- Llévalo en línea
 - Con un backend se puede hacer el juego hasta más interactivo
- A tu gusto
 - Agregar la habilidad de crear mas cartas personalizadas

Referencias

- Github Repo: <https://github.com/lrnzbr/NSBuilders-Loteria>
- SwiftUI: <https://youtu.be/b1oC7sLlgpl?si=Vg6XledVHHNxsvaW>
- Draggable : <https://youtu.be/lsXqJKm4l-U?si=S6j3AvDWLmvTzHab>
- Scroll Animations :<https://youtu.be/IwUp2iPOjnI?si=tidjFAy8e2zddL5n>

¡Gracias!



GitHub: @Lrnzbr