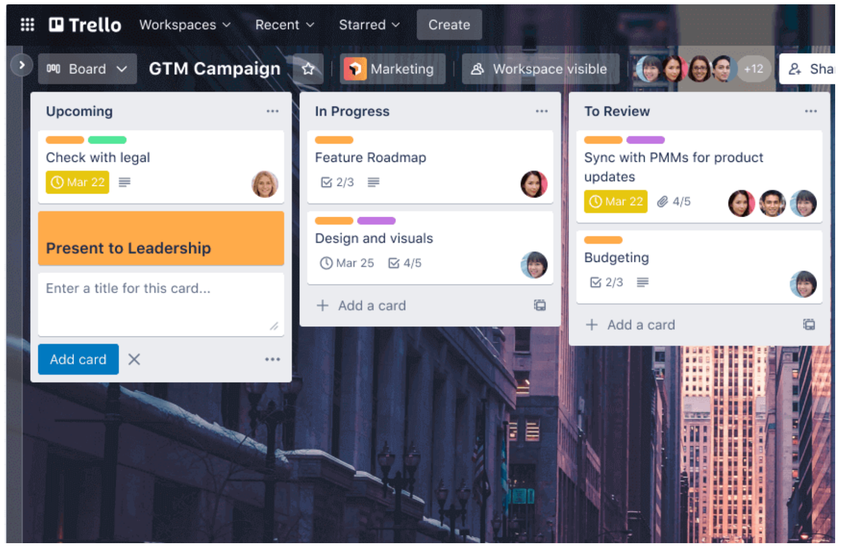
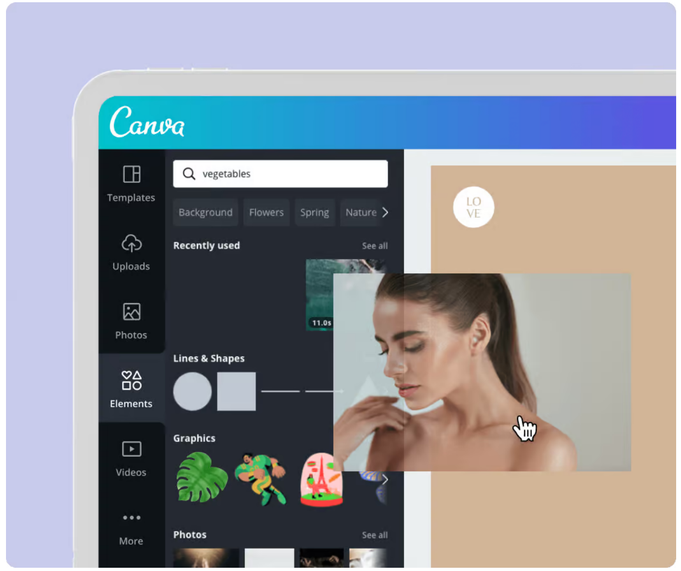
Drag & Drop

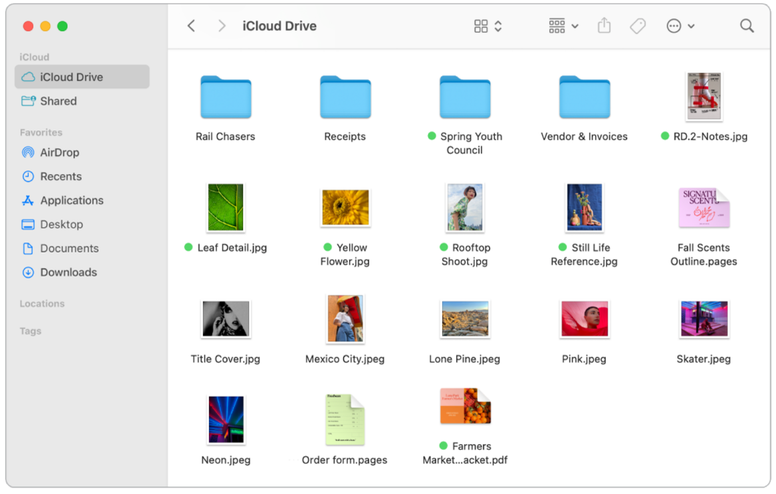
Taller SwiftUI - Semana 18

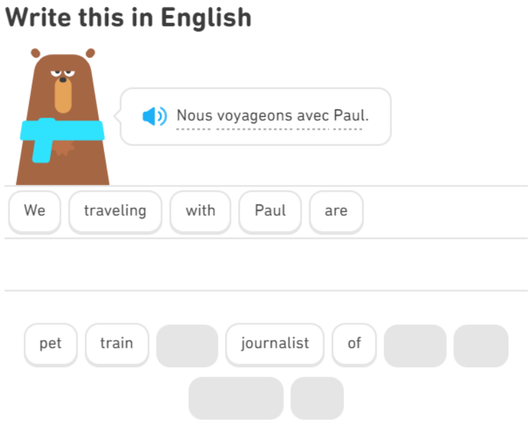
Sergio González - Diciembre 3, 2024

Ejemplos de Drag & Drop

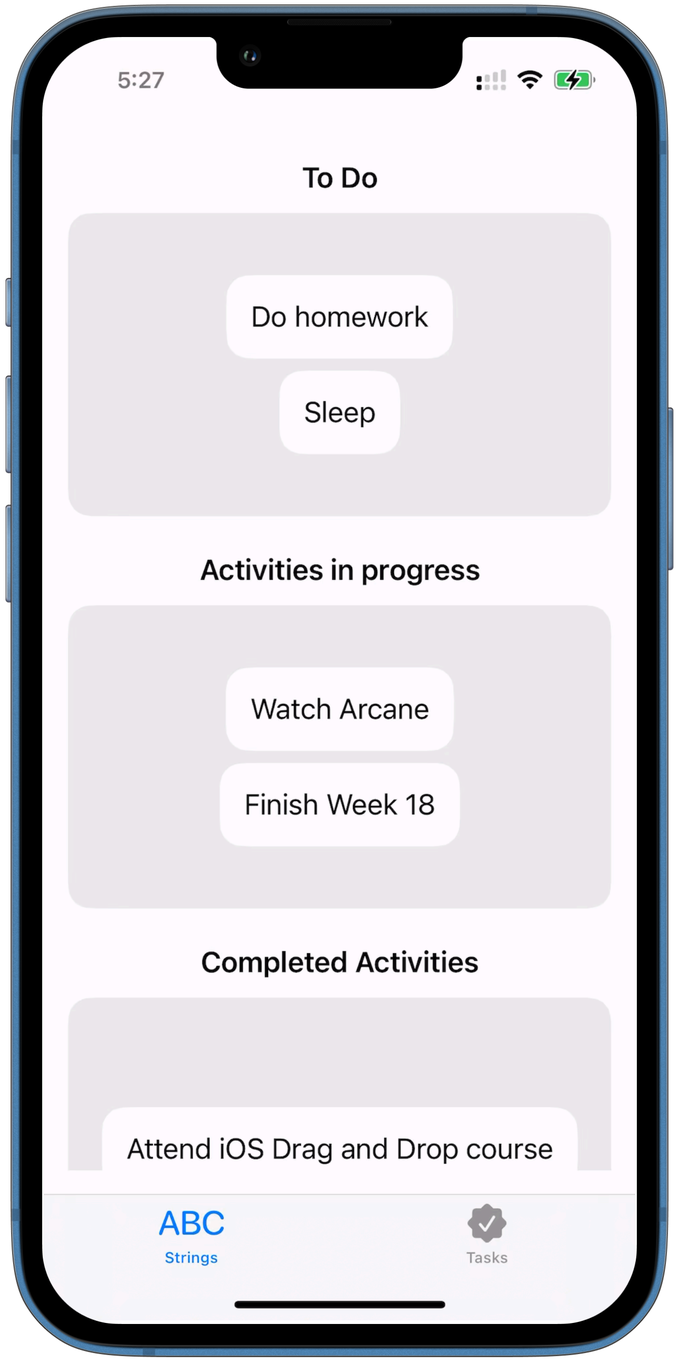








Funcionalidad final



Modificador - draggable

Instance Method

draggable(_:)

Activates this view as the source of a drag and drop operation.

iOS 16.0+ | iPadOS 16.0+ | Mac Catalyst 16.0+ | macOS 13.0+ | visionOS 1.0+

```
nonisolated
func draggable<T>(_ payload: @autoclosure @escaping () -> T) -> some View where T : Transferable
```

Parameters

payload
A closure that returns a single instance or a value conforming to `Transferable` that represents the draggable data from this view.

Modificador - draggable

```
var word: String = "Hello"

var body: some View {
    VStack {
        Text(word)
    }
    .draggable(word)
}
```

Modificador - dropDestination

dropDestination(for:action:isTargeted:)

Defines the destination of a drag and drop operation that handles the dropped content with a closure that you specify.

iOS 16.0+ | iPadOS 16.0+ | Mac Catalyst 16.0+ | macOS 13.0+ | visionOS 1.0+

```
nonisolated
func dropDestination<T>(
    for payloadType: T.Type = T.self,
    action: @escaping ([T], CGPoint) -> Bool,
    isTargeted: @escaping (Bool) -> Void = { _ in }
) -> some View where T : Transferable
```

Parameters

`payloadType`
The expected type of the dropped models.

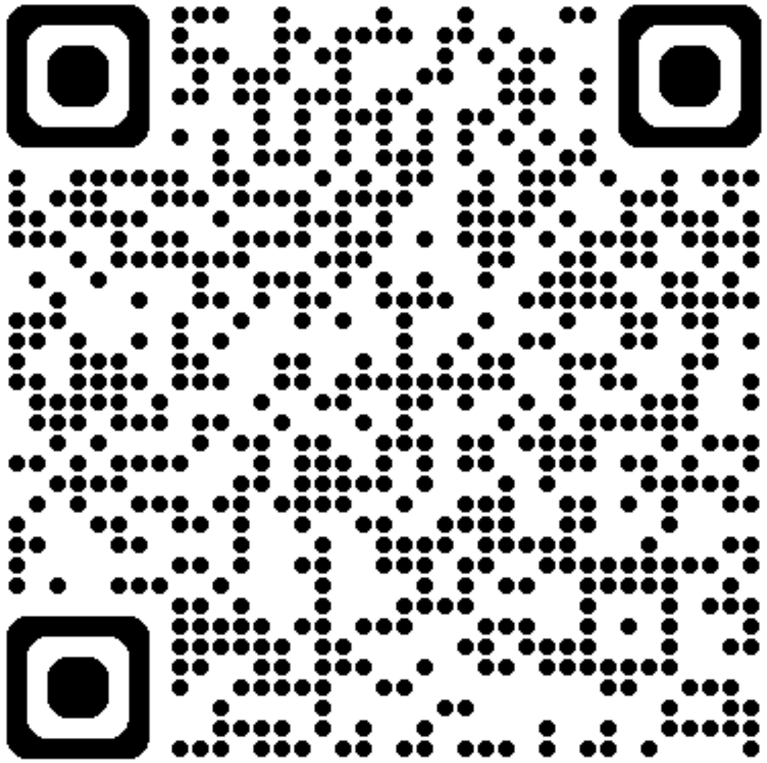
`action`
A closure that takes the dropped content and responds appropriately. The first parameter to `action` contains the dropped items. The second parameter contains the drop location in this view's coordinate space. Return `true` if the drop operation was successful; otherwise, return `false`.

`isTargeted`
A closure that is called when a drag and drop operation enters or exits the drop target area. The received value is `true` when the cursor is inside the area, and `false` when the cursor is outside.

Modificador - dropDestination

```
var body: some View {
    VStack {
        Text("Drop me!")
    }
    .dropDestination(for: String.self) { words, location in
        for word in words {
            print("The word \(word) was dropped into me")
        }
        return true
    } isTargeted: { isTargeted in
        if isTargeted {
            print("Something is hovering over me!")
        } else {
            print("Nothing is happening :(")
        }
    }
}
```

Código Base



Drag and drop con Strings

D&D con String

```
32  struct RowView: View {
46      var body: some View {
47          VStack {
48              Text(rowName)
49                  .bold()
50              ZStack {
51                  RoundedRectangle(cornerRadius: 15)
52                      .frame(maxWidth: .infinity)
53                      .frame(height: 200)
54                      .foregroundStyle(backgroundColor)
55                      .padding(.horizontal)
56
57                  VStack {
58                      ForEach (activities[listIndex], id: \.self) { act in
59                          Text(act)
60                              .padding()
61                              .background(Color(uiColor: .secondarySystemGroupedBackground))
62                              .clipShape(RoundedRectangle(cornerRadius: 15))
63                              .draggable(act) // MARK: Draggable
64                      }
65                  }
66                  // MARK: Easy animation
67                  .animation(.spring, value: activities[listIndex])
68              }
69          }
```

KanbanView.swift

```
70          // MARK: Drop Destination
71          .dropDestination(for: String.self) { acts, location in
72              for i in 0..<activities.count {
73                  activities[i].removeAll(where: { acts.contains($0)} )
74              }
75
76              activities[listIndex].append(contentsOf: acts)
77              return true
78          } isTargeted: { isTargeted in
79              self.isTargeted = isTargeted
80          }
81      }
82  }
83
```

Drag and drop con tipos personalizados

Protocolo Transferable

Define cómo interactúa el objecto con Drag and Drop o Copy and Paste

Task.swift

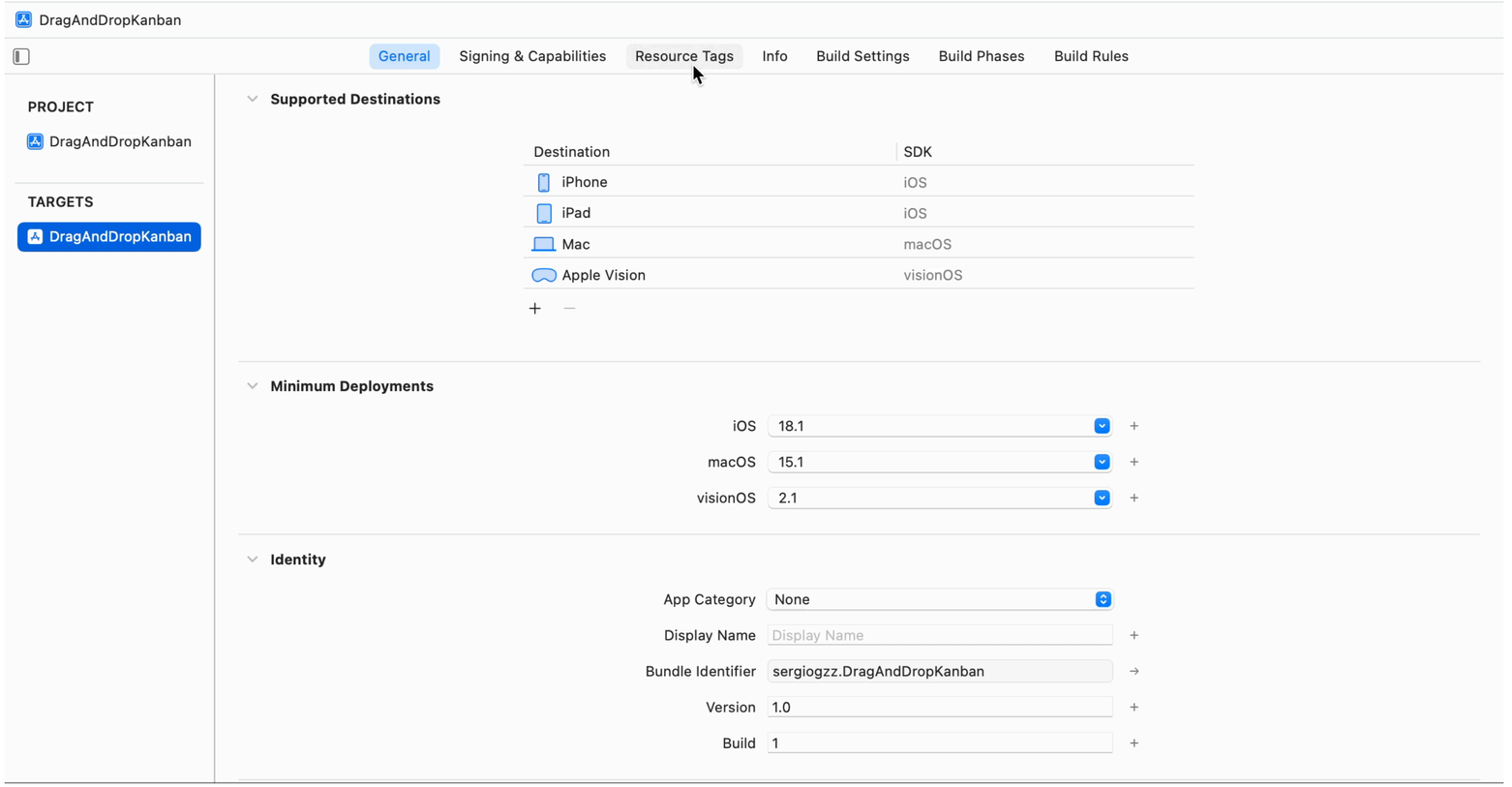
```swift
// MARK: Important imports
import SwiftUI
import UniformTypeIdentifiers
```

```swift
// MARK: - Transferable

// Define a custom UTType for Task
extension UTType {
    static var task: UTType {
        // BundleIdentifier + "." + typeName
        UTType(exportedAs: "sergiogzz.DragAndDropKanban.task")
    }
}


extension Task: Transferable {
    static var transferRepresentation: some TransferRepresentation {
        CodableRepresentation(contentType: .task)
    }
}
```

Protocolo Transferable

DragAndDropKanban.xcodeproj

Se tiene que crear un identificador de tipo exportado



D&D con Tipo Task

```swift
struct RowViewTask: View {
    var body: some View {
        VStack {
            Text(status.rawValue)
                .bold()
            ZStack {
                RoundedRectangle(cornerRadius: 15)
                    .frame(maxWidth: .infinity)
                    .frame(height: 200)
                    .foregroundStyle(backgroundColor)
                    .padding(.horizontal)

                VStack {
                    ForEach (filteredTasks, id: \.id) { task in
                        Text(task.name)
                            .padding()
                            .background(Color(uiColor: .secondarySystemGroupedBackground))
                            .clipShape(RoundedRectangle(cornerRadius: 15))
                            .draggable(task) // MARK: Draggable
                    }
                }
                // MARK: Easy animation
                .animation(.spring, value: tasks)
            }
        }
        // MARK: Drop Destination
        .dropDestination(for: Task.self) { tasks, location in
            for task in tasks {
                guard let referenceTask = self.tasks.first(where: { $0.id == task.id } ) else {
                    continue
                }

                withAnimation {
                    referenceTask.status = self.status
                }
            }
            return true
        } isTargeted: { isTargeted in
            self.isTargeted = isTargeted
        }
    }
}
```

KanbanViewTask.swift

Código Final