

Sesion 04:

**Xcode, Swift, SwiftUI e iOS: Gestión dinámica de listas y
TabView**

Grupos de Trabajo

Líneas de Productos Software

Pablo Gómez Rivas

Departamento de Informática
Universidad de Almería

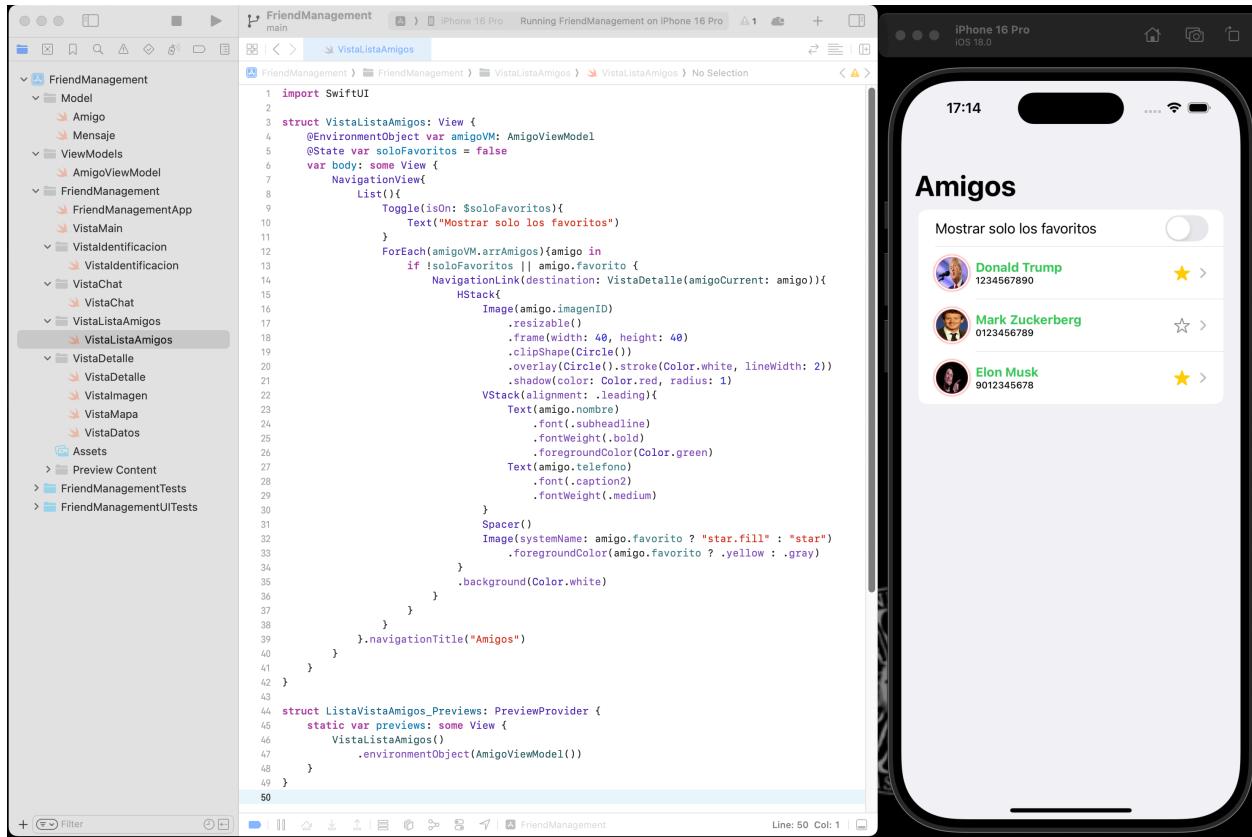
Almería, martes 15 de octubre de 2024

Índice de Contenidos

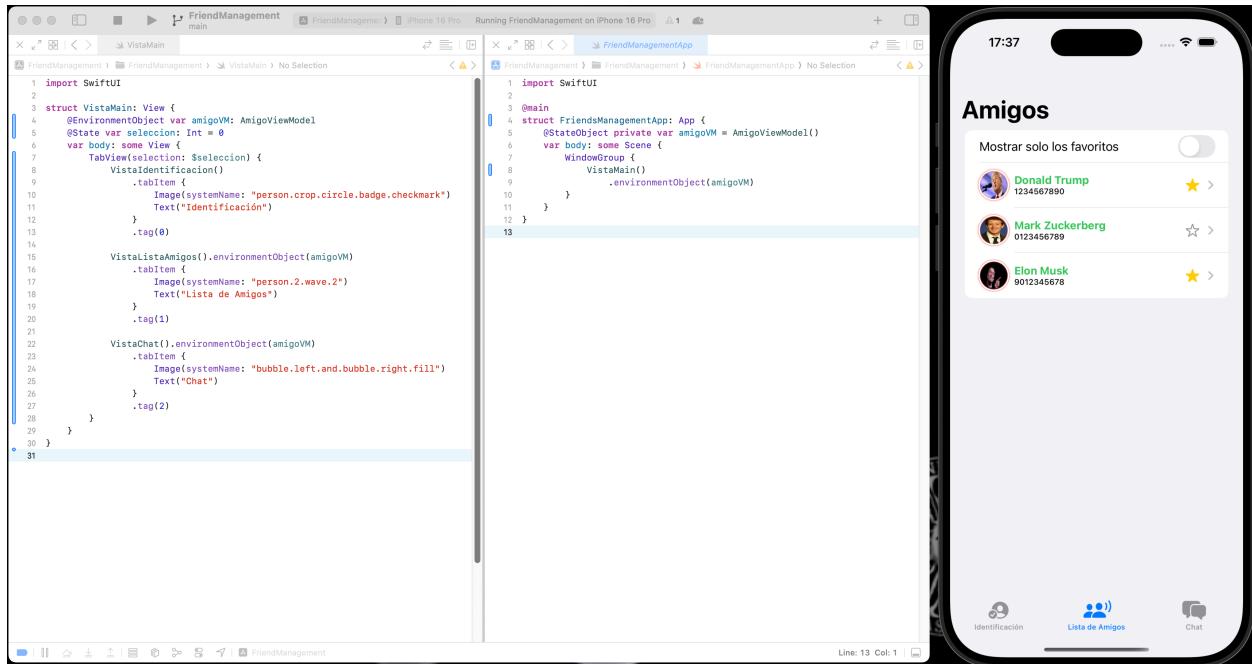
Página

Organizando el proyecto	3
Modificando VistaMain	3
Modificando VistaIdentificacion	4
Ejercicio 1	4
Modificando VistaChat	5
Modificando VistaListaAmigos	5
Paso 1	5
Ejercicio 2	6
Paso 2	7
Paso 3	9
Ejercicio 3	11
Paso 3	12
Modificando VistaAddAmigo	13
Terminando VistaListaAmigos	14
Ejercicio 4	15

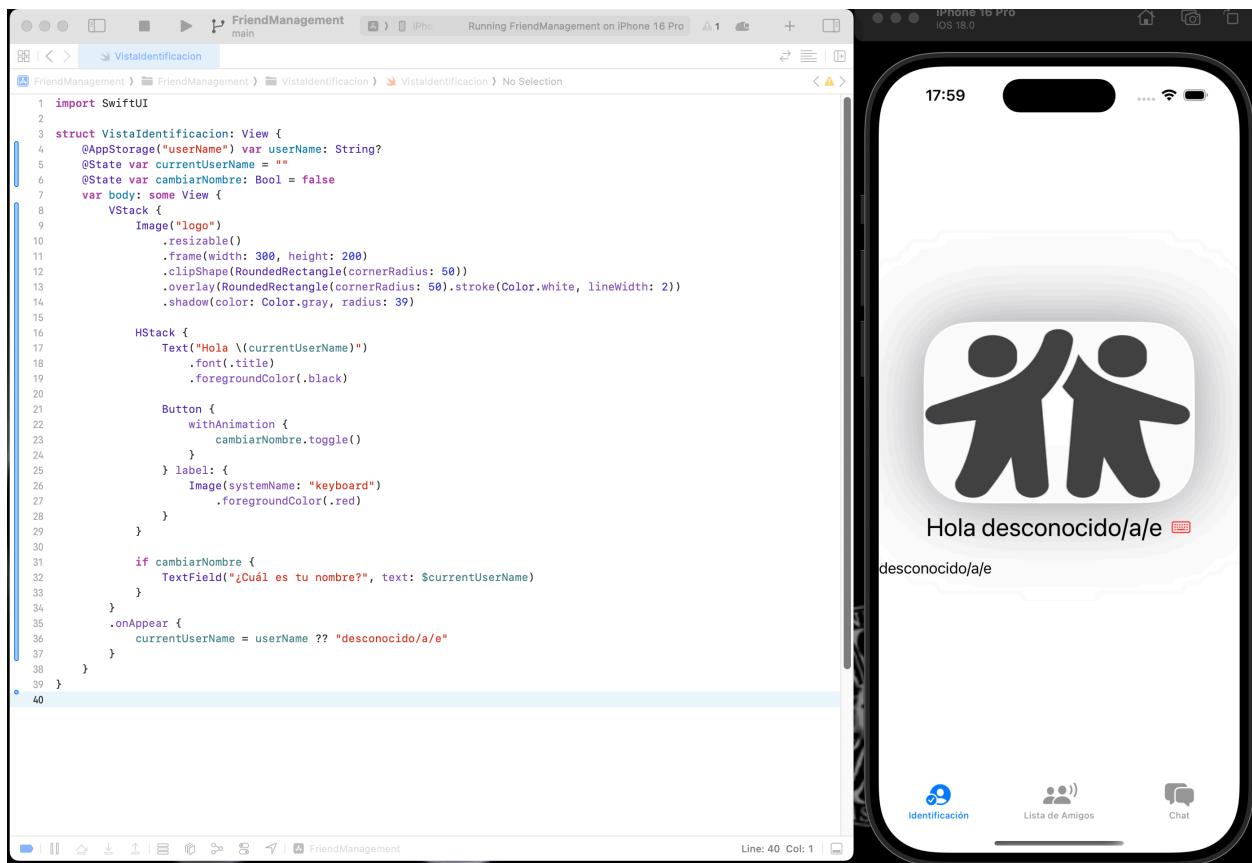
Organizando el proyecto



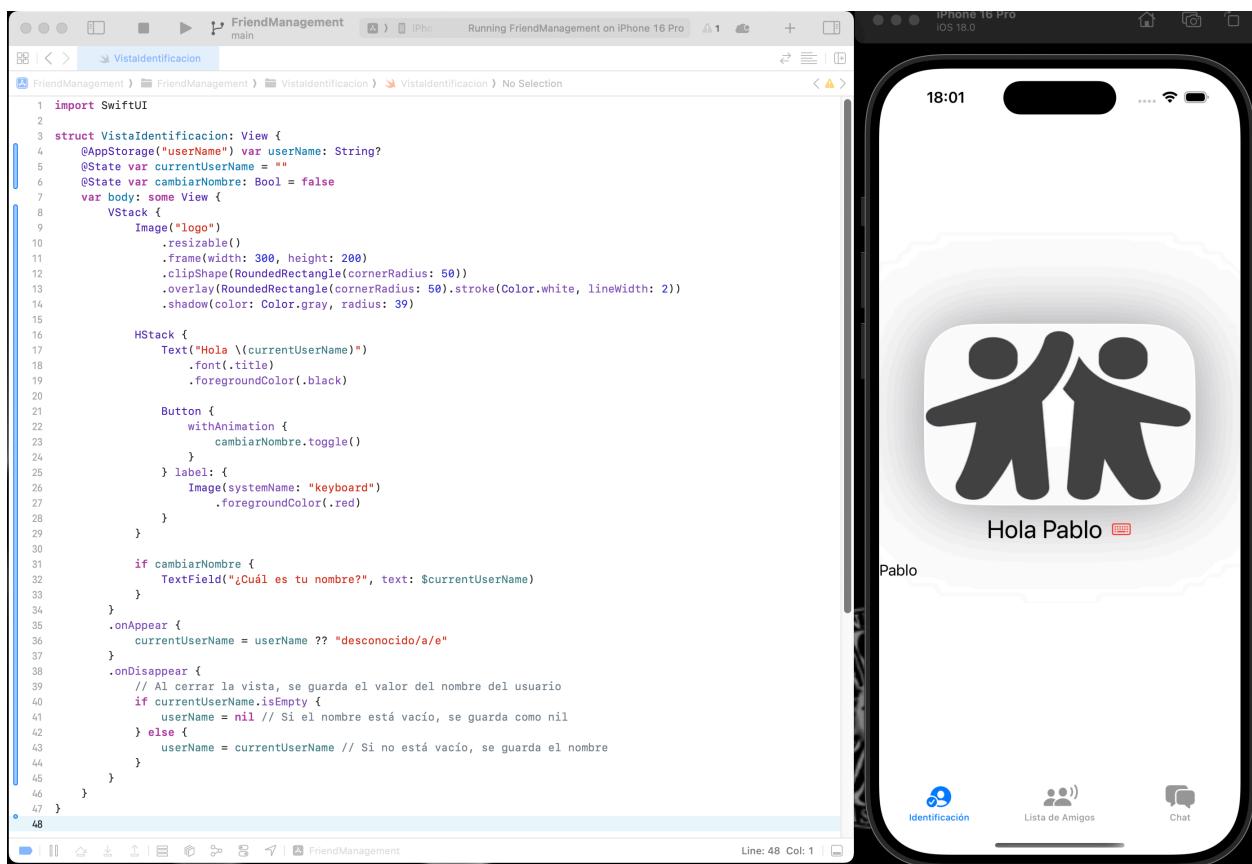
Modificando VistaMain



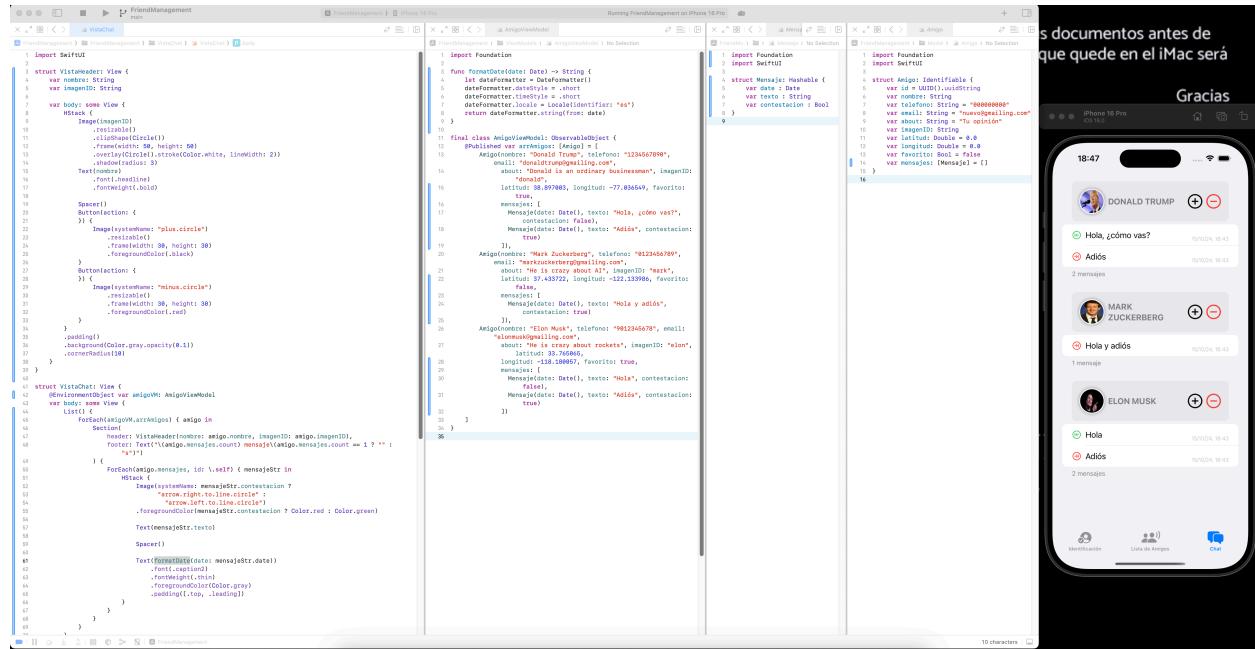
Modificando VistaIdentificacion



Ejercicio 1



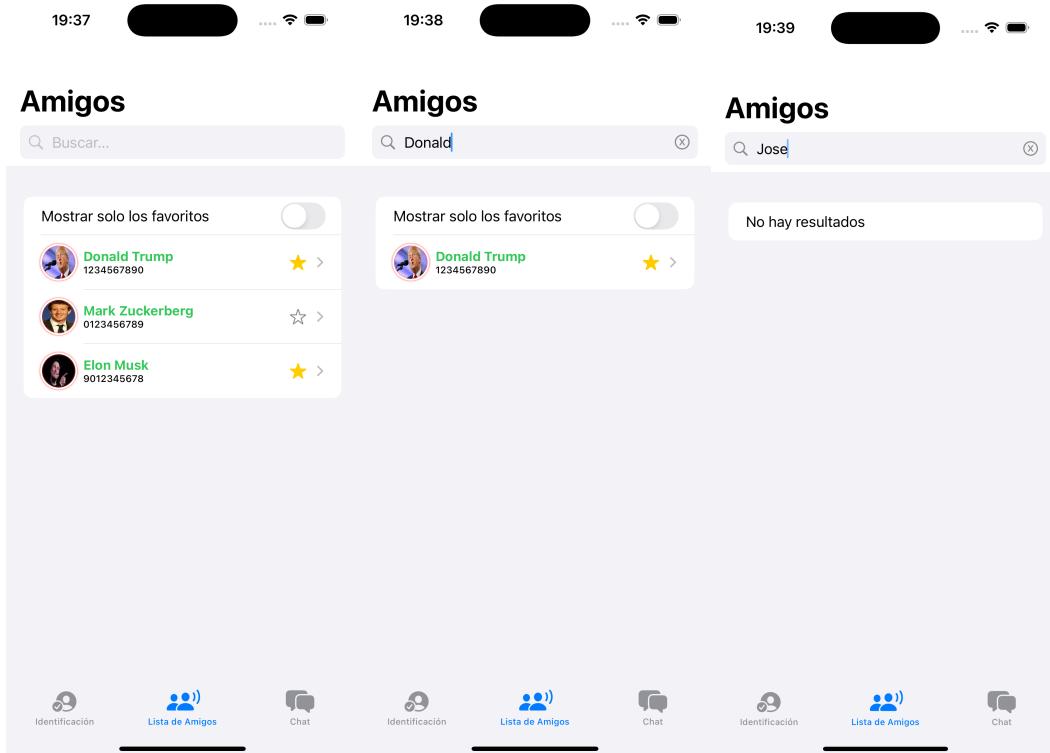
Modificando VistaChat



Modificando VistaListaAmigos

Paso 1





Ejercicio 2

- **Prueba de Búsqueda Vacía:** Se verificó que, al dejar el campo de búsqueda vacío, se muestren todos los amigos. Esto garantiza que el componente de búsqueda no limita la visualización de amigos si no hay un criterio de búsqueda.
- **Prueba de Búsqueda por Nombre:** Se ingresaron diferentes nombres (tanto en mayúsculas como en minúsculas) y se comprobó que solo se mostraban los amigos cuyo nombre contenía la cadena ingresada.
- **Prueba de Favoritos:** Se activó y desactivó la opción "Mostrar solo los favoritos" para asegurarse de que los resultados reflejaran correctamente esta opción. Se verificó que los amigos que no eran favoritos no aparecieran cuando la opción estaba activada.
- **Interacción con el Botón de Limpiar:** Se comprobó que al presionar el botón de limpiar (la 'X'), el campo de búsqueda se vaciaba correctamente, y todos los amigos volvían a aparecer.
- **Pruebas de Interfaz:** Se realizaron pruebas de interfaz en diferentes dispositivos y orientaciones (horizontal y vertical) para verificar que el diseño se mantuviera bien organizado y accesible.

Paso 2

FriendManagement

VistaListaAmigos

```
FriendManagement VistaListaAmigos main FriendManagement iPhone 16 Pro Running FriendManagement on iPhone 16 Pro 1
```

```
FriendManagement FriendManagement VistaListaAmigos VistaListaAmigos body
```

```
3 struct BusquedaView: View {
4     var body: some View {
5         .foregroundColor(Color(UIColor.gray).opacity(0.9))
6     }
7 }
8
9 }
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28 struct VistaListaAmigos: View {
29     @EnvironmentObject var amigoVM: AmigoViewModel
30     @State var query: String = ""
31     @State var soloFavoritos = false
32     var body: some View {
33         NavigationView{
34             VStack {
35                 BusquedaView(text: $query)
36                 let result = amigoVM.arrAmigos.filter { amigo in
37                     (isoloFavoritos || amigo.favorito) &&
38                     (query.isEmpty || amigo.nombre.lowercased().contains(query.lowercased())))
39             }
40             List(){
41                 Toggle(isOn: $soloFavoritos){
42                     Text("Mostrar solo los favoritos")
43                 }
44                 if result.isEmpty {
45                     Text("No hay resultados")
46                 }
47                 ForEach(result) { amigo in
48                     NavigationLink(destination: VistaDetalle(amigoCurrent: amigo)){
49                         HStack{
50                             Image(amigo.imageID)
51                             .resizable()
52                             .frame(width: 40, height: 40)
53                             .clipShape(Circle())
54                             .overlay(Circle().stroke(Color.white, lineWidth:
55                                 2))
56                             .shadow(color: Color.red, radius: 1)
57                         VStack(alignment: .leading){
58                             Text(amigo.nombre)
59                             .font(.subheadline)
60                             .fontWeight(.bold)
61                             .foregroundColor(Color.green)
62                             Text(amigo.telefono)
63                             .font(.caption2)
64                             .fontWeight(.medium)
65                         }
66                         Spacer()
67                         Image(systemName: amigo.favorito ? "star.fill" :
68                               "star")
69                         .foregroundColor(amigo.favorito ? .yellow :
70                               .gray)
71                     }
72                     .background(Color.white)
73                 }
74             }
75         }.navigationTitle("Amigos")
76     }
77 }
```

```
FriendManagement FriendManagement VistaDatos VistaDatos No Selection
```

```
1 import SwiftUI
2
3 struct VistaDatos: View {
4     @EnvironmentObject private var amigoVM: AmigoViewModel
5     @State private var textoOpinion: String = ""
6     @State private var favorito = false
7     @State private var amigoIndex: Int = -1
8     var amigoCurrent: Amigo
9     var body: some View {
10         VStack {
11             HStack{
12                 Text(amigoCurrent.nombre)
13                 .font(.title)
14                 .foregroundColor(.white)
15                 Button{
16                     favorito.toggle()
17                     label: {
18                         Image(systemName: favorito ? "star.fill" : "star")
19                         .foregroundColor(favorito ? .yellow : .gray)
20                     }
21                 }
22             }
23             Label(amigoCurrent.telefono, systemImage:"iphone")
24             .font(.body)
25             Link(destination: URL(string: "mailto:" + amigoCurrent.email)!, label: {
26                 Image(systemName: "livephoto")
27                 .frame(width: 20, height: 20, alignment: .center)
28                 Text(amigoCurrent.email)
29             })
30             Divider()
31             HStack {
32                 Text("About " + amigoCurrent.nombre)
33                 .font(.headline)
34                 Image(systemName: "pencil")
35                 .foregroundColor(.white)
36                 .font(.headline)
37             }
38             TextEditor(text: $textoOpinion)
39             .frame(width: 350, height: 200)
40             .overlay(Rectangle().stroke(Color.black, lineWidth: 1))
41             .onChange(of: textoOpinion) {
42                 if (textoOpinion.count > 150) {
43                     textoOpinion = String(textoOpinion.prefix(150))
44                 }
45             }
46             .font(.footnote)
47             .scrollContentBackground(.hidden)
48             let characterCount: Int = Int(textoOpinion.count)
49             ProgressView("Número de caracteres: \(characterCount)/150", value:
50             min(Double(characterCount), 150), total: 150)
51             .progressViewStyle(LinearProgressViewStyle())
52             .accentColor(characterCount >= 100 ? .red : .blue)
53             .frame(width: 300)
54             Spacer()
55         }.background(Color.brown)
56         .onAppear {
57             textoOpinion = amigoCurrent.about
58             favorito = amigoCurrent.favorito
59             amigoIndex = amigoVM.arrAmigos.firstIndex(where: {$0.id ==
60                 amigoCurrent.id})!
61         }
62         .onDisappear {
63             amigoVM.arrAmigos[amigoIndex].about = textoOpinion
64             amigoVM.arrAmigos[amigoIndex].favorito = favorito
65         }
66     }
67 }
```

12:31

12:32

Amigos

Buscar...

Mostrar solo los favoritos

Nombre	Teléfono	Acciones
Donald Trump	234567890	>
Mark Zuckerberg	0123456789	>
Elon Musk	9012345678	>

Amigos

Buscar...

Mostrar solo los favoritos

Nombre	Teléfono	Acciones
Elon Musk	9012345678	>

Identificación Lista de Amigos Chat

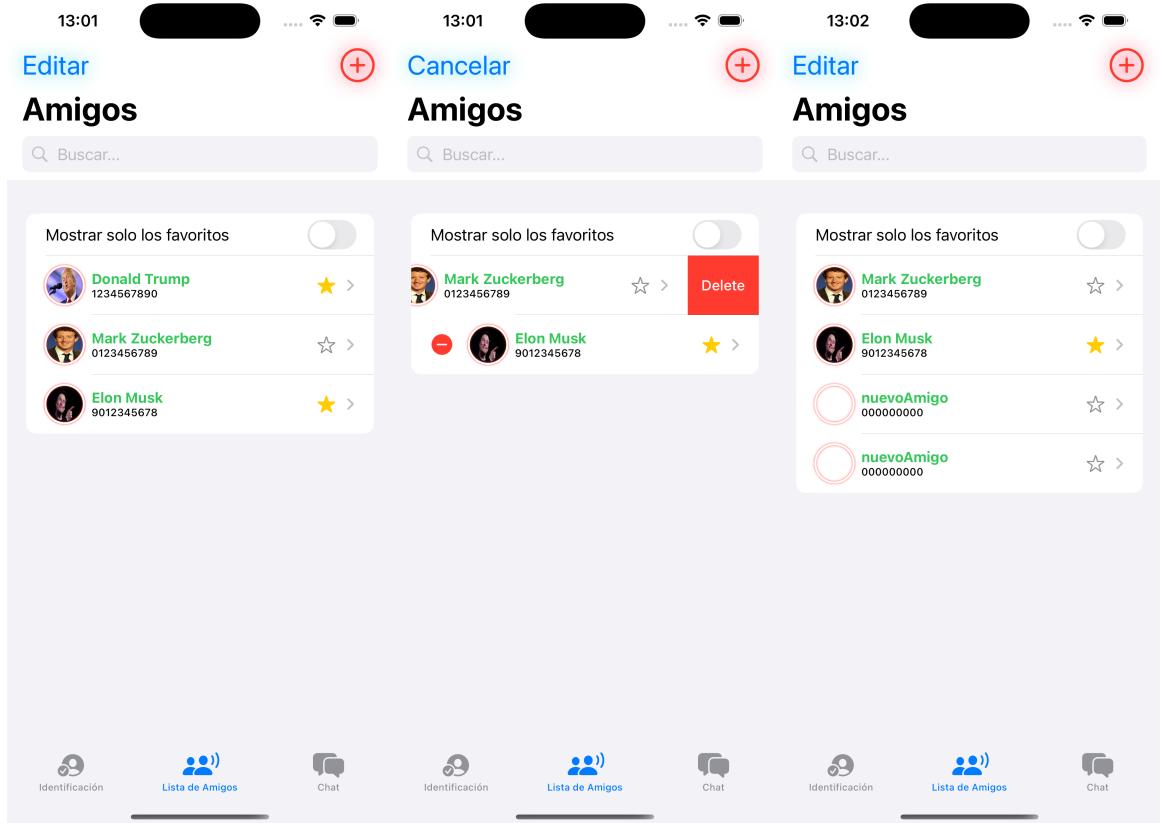
Identificación Lista de Amigos Chat

Paso 3

```

28 struct VistaListaAmigos: View {
29     @State var soloFavoritos = false
30     @State var enEdicion: Bool = false
31     var body: some View {
32         NavigationView{
33             VStack {
34                 BusquedaView(text: $query)
35                 let result = amigoVM.arrAmigos.filter { amigo in
36                     (!soloFavoritos || amigo.favorito) &&
37                     (query.isEmpty || amigo.nombre.lowercased().contains(query.lowercased())))
38                 }
39             }
40             List(){
41                 Toggle(isOn: $soloFavoritos){
42                     Text("Mostrar solo los favoritos")
43                 }
44                 if result.isEmpty {
45                     Text("No hay resultados")
46                 }
47                 ForEach(result) { amigo in
48                     NavigationLink(destination: VistaDetalle(amigoCurrent: amigo)){
49                         HStack{
50                             Image(amigo.imagenID)
51                                 .resizable()
52                                 .frame(width: 40, height: 40)
53                                 .clipShape(Circle())
54                                 .overlay(Circle().stroke(Color.white, lineWidth: 2))
55                                 .shadow(color: Color.red, radius: 1)
56                         VStack(alignment: .leading){
57                             Text(amigo.nombre)
58                                 .font(.subheadline)
59                                 .fontWeight(.bold)
60                                 .foregroundColor(Color.green)
61                             Text(amigo.telefono)
62                                 .font(.caption2)
63                                 .fontWeight(.medium)
64                         }
65                         Spacer()
66                         Image(systemName: amigo.favorito ? "star.fill" : "star")
67                             .foregroundColor(amigo.favorito ? .yellow : .gray)
68                     }
69                     .background(Color.white)
70                 }
71             }
72             .onDelete { indexSet in amigoVM.arrAmigos.remove(atOffsets: indexSet) }
73         }.navigationTitle("Amigos")
74         .navigationBarItems(
75             leading:
76                 Button(){
77                     enEdicion.toggle()
78                 }
79             label: {
80                 Text(enEdicion ? "Cancelar" : "Editar")
81                     .font(.title)
82                     .foregroundColor(Color.blue)
83                     .shadow(color: Color(red: 0.28, green: 0.855, blue: 0.92), radius:9)
84             },
85             trailing:
86                 Button(){
87                     amigoVM.arrAmigos.append(
88                         Amigo(nombre: "nuevoAmigo", imagenID: "person"))
89                 }
90             label: {
91                 Image(systemName: "plus.circle")
92                     .font(.title)
93                     .foregroundColor(Color.red)
94                     .shadow(color: Color.pink, radius:9)
95             })
96             .environment(\.EditMode, .constant(enEdicion ? EditMode.active : EditMode.inactive))
97         }
98     }
99 }

```



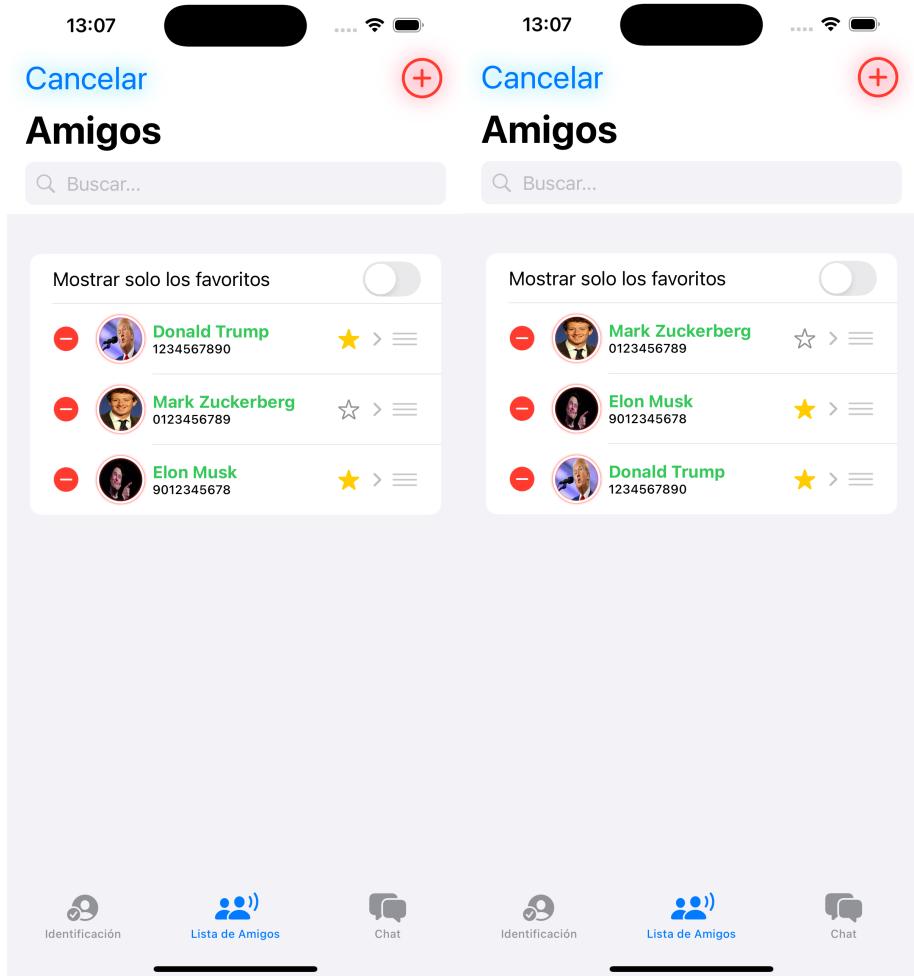
Ejercicio 3



```

FriendManagement
main
FriendManagement > FriendManagement > VistaListaAmigos > VistaListaAmigos > body
28 struct VistaListaAmigos: View {
29     var body: some View {
30         BusquedaView(text: $query)
31         let result = amigoVM.arrAmigos.filter { amigo in
32             (!soloFavoritos || amigo.favorito) &&
33             (query.isEmpty || amigo.nombre.lowercased().contains(query.lowercased())))
34         }
35     }
36     List(){
37         Toggle(isOn: $soloFavoritos){
38             Text("Mostrar solo los favoritos")
39         }
40         if result.isEmpty {
41             Text("No hay resultados")
42         }
43         ForEach(result) { amigo in
44             NavigationLink(destination: VistaDetalle(amigoCurrent: amigo)){
45                 HStack{
46                     Image(amigo.imagenID)
47                         .resizable()
48                         .frame(width: 40, height: 40)
49                         .clipShape(Circle())
50                         .overlay(Circle().stroke(Color.white, lineWidth: 2))
51                         .shadow(color: Color.red, radius: 1)
52                     VStack(alignment: .leading){
53                         Text(amigo.nombre)
54                             .font(.subheadline)
55                             .fontWeight(.bold)
56                             .foregroundColor(Color.green)
57                         Text(amigo.telefono)
58                             .font(.caption2)
59                             .fontWeight(.medium)
60                     }
61                     Spacer()
62                     Image(systemName: amigo.favorito ? "star.fill" : "star")
63                         .foregroundColor(amigo.favorito ? .yellow : .gray)
64                 }
65             }
66         }
67         .onDelete { indexSet in amigoVM.arrAmigos.remove(atOffsets: indexSet) }
68         .onMove { indices, newOffset in
69             amigoVM.arrAmigos.move(fromOffsets: indices, toOffset: newOffset)
70         }
71     }
72 }.navigationTitle("Amigos")
73 .navigationBarItems(
74     leading:
75         Button(){
76             enEdicion.toggle()
77         }
78     label:{}
79         Text(enEdicion ? "Cancelar" : "Editar")
80             .font(.title)
81             .foregroundColor(Color.blue)
82             .shadow(color: Color(red: 0.28, green: 0.855, blue: 0.92), radius:9)
83     },
84     trailing:
85         Button(){
86             amigoVM.arrAmigos.append(
87                 Amigo(nombre: "nuevoAmigo", imagenID: "person"))
88         }
89     label: {}
90         Image(systemName: "plus.circle")
91             .font(.title)
92             .foregroundColor(Color.red)
93             .shadow(color: Color.pink, radius:9)
94     })
95     .environment(\.EditMode, .constant(enEdicion ? EditMode.active : EditMode.inactive))
96 }
97
98
99
100
101
102
103

```



Paso 3

FriendManagement

```

1 struct VistaListaAmigos: View {
2     var body: some View {
3         Text("Hello, World!")
4     }
5 }
6
7 struct VistaAddAmigo: View {
8     @Binding var nombre: String
9     @Binding var telefono: String
10    @Binding var email: String
11    @Binding var imagenID: String
12
13    var body: some View {
14        Text("Hello, World!")
15    }
16 }
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108

```

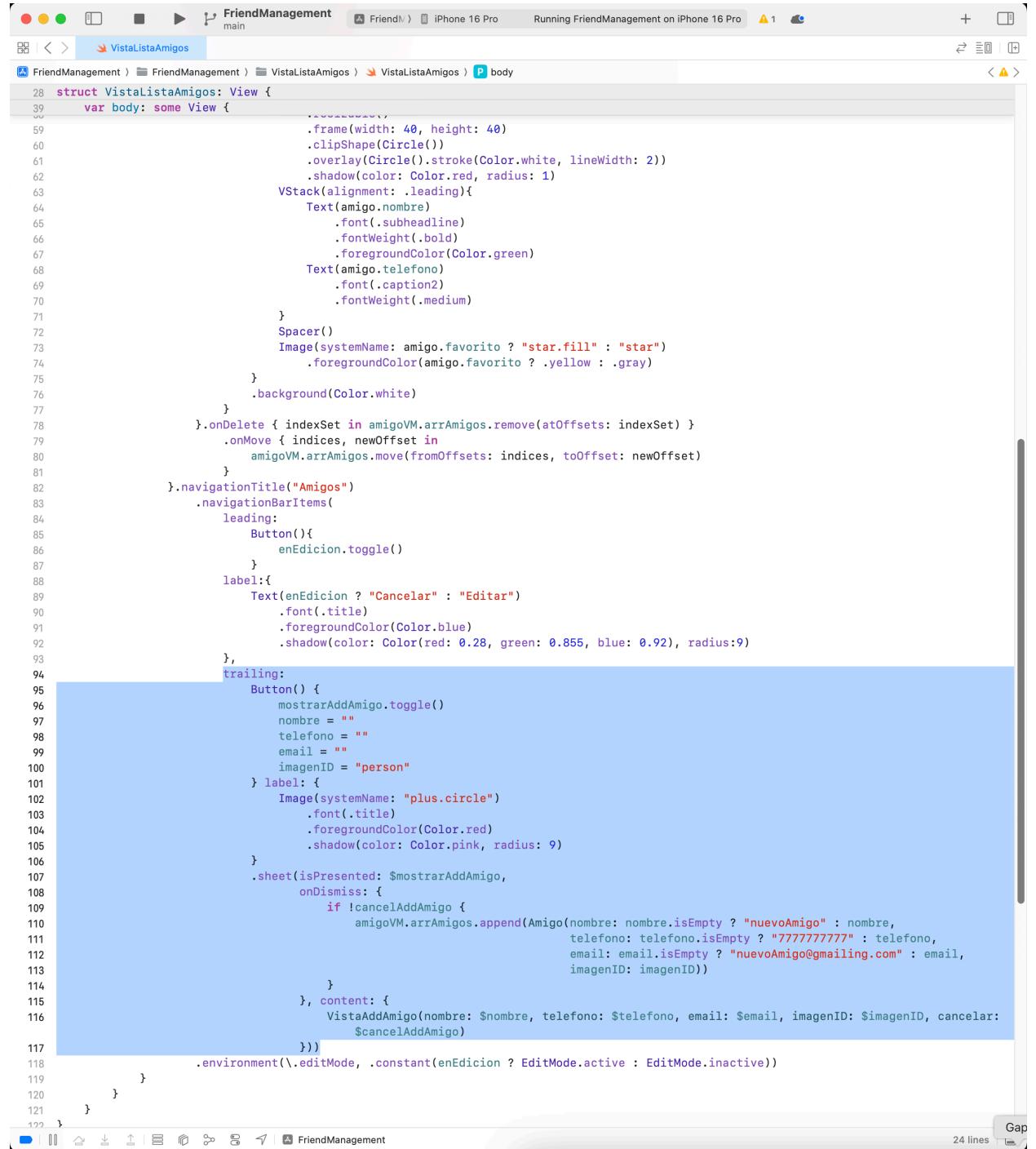
Modificando VistaAddAmigo

```

1 import SwiftUI
2
3 struct VistaAddAmigo: View {
4     @Binding var nombre: String
5     @Binding var telefono: String
6     @Binding var email: String
7     @Binding var imagenID: String
8     @Binding var cancelar: Bool
9     @Environment(\.presentationMode) var modoPresentacion
10    var body: some View {
11        VStack {
12            Form() {
13                Section(header: Text("Fotografía")) {
14                    HStack {
15                        Spacer()
16                        Image(imagenID)
17                            .resizable()
18                            .scaledToFit()
19                            .frame(width: 150, height: 150, alignment: .center)
20                            .clipShape(Circle())
21                            .overlay(Circle().stroke(Color.white, lineWidth: 2))
22                            .shadow(color: Color.gray, radius: 9)
23                    }
24                }
25            }
26            Section(header: Text("Datos")) {
27                TextField("Nombre del amigo", text: $nombre)
28                TextField("Teléfono del amigo", text: $telefono)
29                TextField("Email del amigo", text: $email)
30            }
31        }
32    }
33}
34
35 HStack {
36    Button() {
37        cancelar = true
38        modoPresentacion.wrappedValue.dismiss()
39    } label: {
40        Text("Cancelar")
41            .fontWeight(.bold)
42            .frame(width: 150, height: 50)
43            .foregroundColor(.white)
44            .background(Color.red)
45            .cornerRadius(88)
46    }
47
48    Button() {
49        cancelar = false
50        modoPresentacion.wrappedValue.dismiss()
51    } label: {
52        Text("Guardar")
53            .fontWeight(.bold)
54            .frame(width: 150, height: 50)
55            .foregroundColor(.white)
56            .background(Color.blue)
57            .cornerRadius(88)
58    }
59
60 }
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108

```

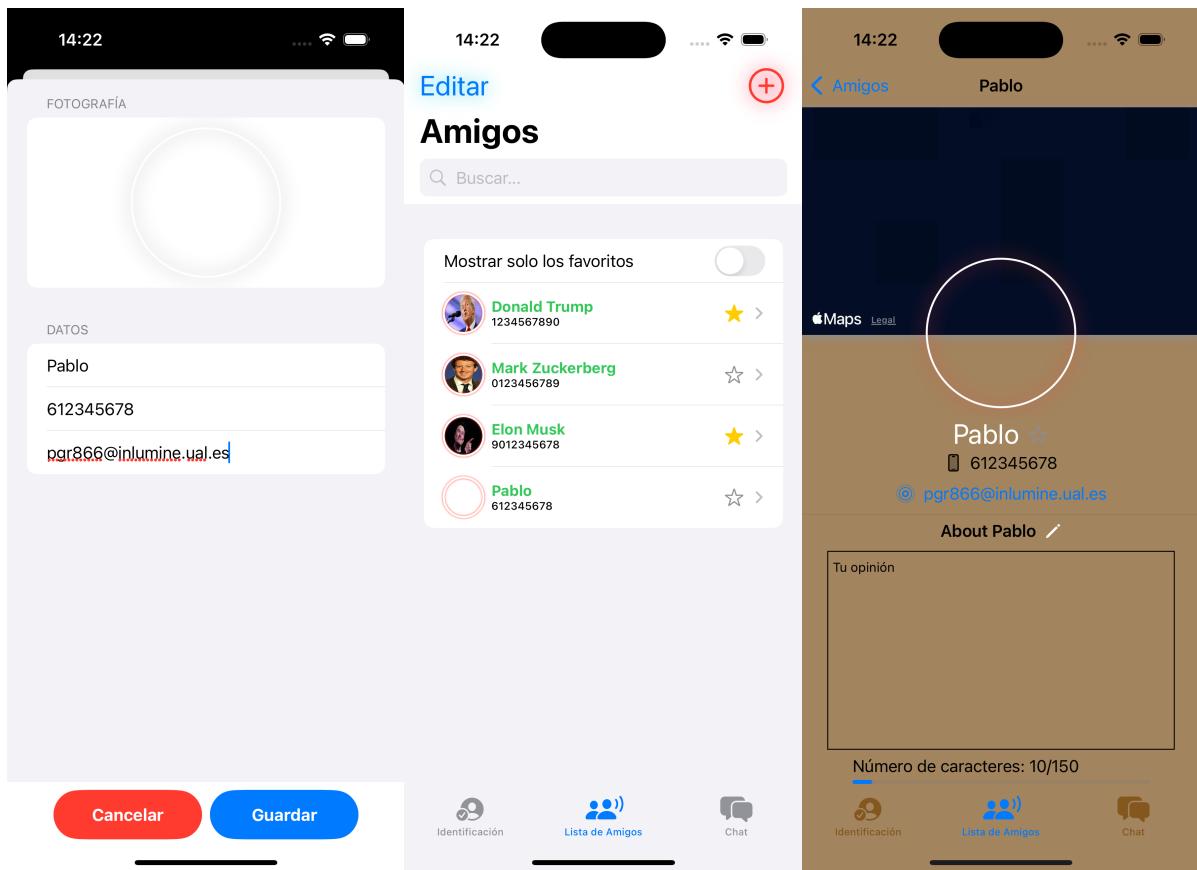
Terminando VistaListaAmigos



```

28 struct VistaListaAmigos: View {
29     var body: some View {
30         .frame(width: 40, height: 40)
31             .clipShape(Circle())
32             .overlay(Circle().stroke(Color.white, lineWidth: 2))
33             .shadow(color: Color.red, radius: 1)
34
35         VStack(alignment: .leading){
36             Text(amigo.nombre)
37                 .font(.subheadline)
38                 .fontWeight(.bold)
39                 .foregroundColor(Color.green)
40             Text(amigo.telefono)
41                 .font(.caption2)
42                 .fontWeight(.medium)
43
44             Spacer()
45             Image(systemName: amigo.favorito ? "star.fill" : "star")
46                 .foregroundColor(amigo.favorito ? .yellow : .gray)
47
48         }
49         .background(Color.white)
50
51     }
52
53     .onDelete { indexSet in amigoVM.arrAmigos.remove(atOffsets: indexSet) }
54     .onMove { indices, newOffset in
55         amigoVM.arrAmigos.move(fromOffsets: indices, toOffset: newOffset)
56     }
57
58     .navigationTitle("Amigos")
59     .navigationBarItems(
60         leading:
61             Button(){
62                 enEdicion.toggle()
63             }
64         label:{
65             Text(enEdicion ? "Cancelar" : "Editar")
66                 .font(.title)
67                 .foregroundColor(Color.blue)
68                 .shadow(color: Color(red: 0.28, green: 0.855, blue: 0.92), radius:9)
69         },
69         trailing:
70             Button()
71                 .mostrarAddAmigo.toggle()
72                 nombre = ""
73                 telefono = ""
74                 email = ""
75                 imagenID = "person"
76
77         label: {
78             Image(systemName: "plus.circle")
79                 .font(.title)
80                 .foregroundColor(Color.red)
81                 .shadow(color: Color.pink, radius: 9)
82         }
83
84         .sheet(isPresented: $mostrarAddAmigo,
85             onDismiss: {
86                 if !cancelAddAmigo {
87                     amigoVM.arrAmigos.append(Amigo(nombre: nombre.isEmpty ? "nuevoAmigo" : nombre,
88                                         telefono: telefono.isEmpty ? "7777777777" : telefono,
89                                         email: email.isEmpty ? "nuevoAmigo@gmailing.com" : email,
90                                         imagenID: imagenID))
91
92                 }
93             }, content: {
94                 VistaAddAmigo(nombre: $nombre, telefono: $telefono, email: $email, imagenID: $imagenID, cancelar: $cancelAddAmigo)
95             })
96
97         .environment(\.EditMode, .constant(enEdicion ? EditMode.active : EditMode.inactive))
98     }
99 }
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122

```



Ejercicio 4

En cuanto a la legibilidad del código, una buena mejora sería documentarlo y comentarlo, para que en caso de que otros desarrolladores tuvieran que mantenerlo esté bien estructurado y organizado.