BOTONES Y GRADIENTES

- Un botón tiene básicamente dos partes:
  1. Que queremos que se vea
  2. Que queremos que haga
- Para crear un borde al botón usaremos el modificador .border
- El cornerRadius aplica solo a los textos pero no a los bordes, no nos redondea los bordes, así que si queremos hacer un botón con borde y que éste sea redondeado tenemos que recuperar igual que en las imágenes el overlay y aplicarle algún modificador.
- Modificador para gradientes = .backgroundColor(LienerGradient….
  Web con ideas para gradientes = **uigradients.com**
  Para crear colores vamos a assets -> botón derecho -> new color set. Seleccionamos el color en el centro y en el panel de la derecha de los atributos, en el inspector, en el name le damos un nombre. Luego en imput Method, en este caso, como la web de uigradients.com están los colores en hexadecimal, seleccionamos esa opción y copiamos el código de color de la web y la o pegamos aquí. Hacemos lo mismo con el color de inicio del degradado y con el color de fin.
- Para dibujar sombras a un botón lo hacemos con el modificador:
  .shadow(radius: 10.0, color: .blue, x:20, y:5)
- Para que el botón ocupe pantalla completa de ancho usamos el frame:
  .frame(minwidht: 0, maxWidth: .infinity)
- Para crear tus propios estilos de botones se usa el Protocolo ButtonStyle, así que crearíamos una estructura que se conforme a este protocolo: por ej: struct BasicButtonStyle: ButtonStyle { …nos dejamos ayudar…, aunque es mejor empezar a escribir la palabra func y elegir la opción makeBody…(ver abajo en el ejemplo). Luego en cada botón al final quitmos los modificadores individuales y llamamos a: .buttonStyle(NombreQueLeHabiamosDadoALaStruct()).
- Animación de botones al pulsarlos:
  .scaleEffect(configuration.isPressed ? 0.8 : 1.0)
- Modificador para rotar algo a “x” grados:
  rotationEffect(Angle(degress: 90.0)

```
import SwiftUI

struct ContentView: View {
   var body: some View {
      VStack {

         Button {
            print("Botón con icono editar pulsado")
         } label: {
            HStack {
               Image(systemName: "square.and.pencil")
               Text("Editar")

            }

         }.buttonStyle(BasicButtonStyle())

         Button {
            print("Botón con icono eliminar pulsado")
         } label: {
            HStack {
               Image(systemName: "trash")
               Text("Eliminar")

            }

         }.buttonStyle(BasicButtonStyle())

         Button {
            print("Botón con icono compartir pulsado")
         } label: {
            HStack {
               Image(systemName: "square.and.arrow.up")
               Text("Compartir")
```

```
            }

        }.buttonStyle(BasicButtonStyle())

        Button {
            print("Botón papelera pulsado")
        } label: {
            Image(systemName: "trash")

        }.buttonStyle(trashButtonStyle())


      }

   }
}

struct ContentView_Previews: PreviewProvider {
   static var previews: some View {
      ContentView()
   }
}

struct BasicButtonStyle: ButtonStyle{
   func makeBody(configuration: Configuration) -> some View {
      configuration.label
         .frame(minWidth:0, maxWidth: .infinity)
         .padding()
         .font(.system(.largeTitle, design: .rounded))
         .foregroundColor(.white)
         //.background(Color(red: 90/255, green: 90/255, blue: 90/255))
         //.clipShape(Circle())
         //.background(LinearGradient(gradient: Gradient(colors: [.purple, .gray]), startPoint: .top, endPoint: .bottom))
         .background(LinearGradient(colors: [Color("Combi 1"), Color("Combi 2"), Color("Combi 3")], startPoint: .leading,
endPoint: .trailing))
         .cornerRadius(45)
         .shadow(color: .init(red: 153/255, green: 153/255, blue: 0/255), radius: 20.0, x: 20, y: 10)
         .padding(.horizontal, 15)
         .scaleEffect(configuration.isPressed ? 0.7 : 1.0)
         //.rotationEffect(Angle(degrees: 90.0)) //ROTAR ALGO 90 GRADOS.
         //.rotationEffect(configuration.isPressed ? (Angle(degrees: 90.0)) : (Angle(degrees: 0.0)))
   }
}

struct trashButtonStyle: ButtonStyle {
   func makeBody(configuration: Configuration) -> some View {
      configuration.label
      .padding()
      .font(.largeTitle)
      .background(Color.red)
      .foregroundColor(.white)
      .clipShape(Circle())
      .rotationEffect(configuration.isPressed ? Angle(degrees: 90.0) : Angle(degrees: 0.0))

   }
}
```