



**INSTITUTO TECNOLÓGICO DE LA LAGUNA**

**DESARROLLO EN iOS**

**APLICACIÓN: signALL**

**Arel Jair Monreal Luna**

**No. Control: 16130827**

**Fecha: 23 de Julio del 2020**

# Índice

Introducción.....	3
Objetivo .....	3
Descripción del proyecto.....	4
Diseño de app y navegación.....	4
<i>Navegación</i> .....	4
<i>Diseño</i> .....	4
Características funcionales .....	6
Código .....	7
Resultados .....	15
Anexos .....	17

## **Introducción**

el desarrollo de aplicaciones móviles son los procedimientos y procesos establecidos que intervienen cuando se crea software para pequeños dispositivos informáticos inalámbricos, como tabletas y teléfonos inteligentes. Al igual que el desarrollo de aplicaciones web, los procesos de desarrollo de aplicaciones móviles tienen sus raíces en el desarrollo de software tradicional.

Cuando se trata del desarrollo de aplicaciones móviles, uno requiere acceso a kits de desarrollo de software (SDK) que permiten a los programadores diseñar y probar su aplicación de código en un entorno simulado controlado.

Los SDK comúnmente utilizados son:

Unity

Android SDK

Licencia de Desarrollador iOS (necesaria para desarrollar aplicaciones para iOS)

Para crear una aplicación móvil exitosa, los programadores deben pasar por las siguientes fases:

La investigación: Refinación de la idea a través de la investigación

Wireframing: creación del marco esquelético de la aplicación

Evaluación de viabilidad técnica: teniendo en cuenta los sistemas de back-end de la aplicación

Prototipo: un prototipo rápido es el concepto de la aplicación en realidad

Diseño: incluye codificación y diseño de la interfaz

Desarrollo: desarrollo progresivo de la aplicación

Pruebas: prueba de funcionalidad y cualquier error que deba corregirse.

Implementación: presentación de la aplicación móvil final.

## **Objetivo**

Desarrollar una aplicación en SwiftUI la cual pueda ser utilizada como herramienta de aprendizaje para niños.

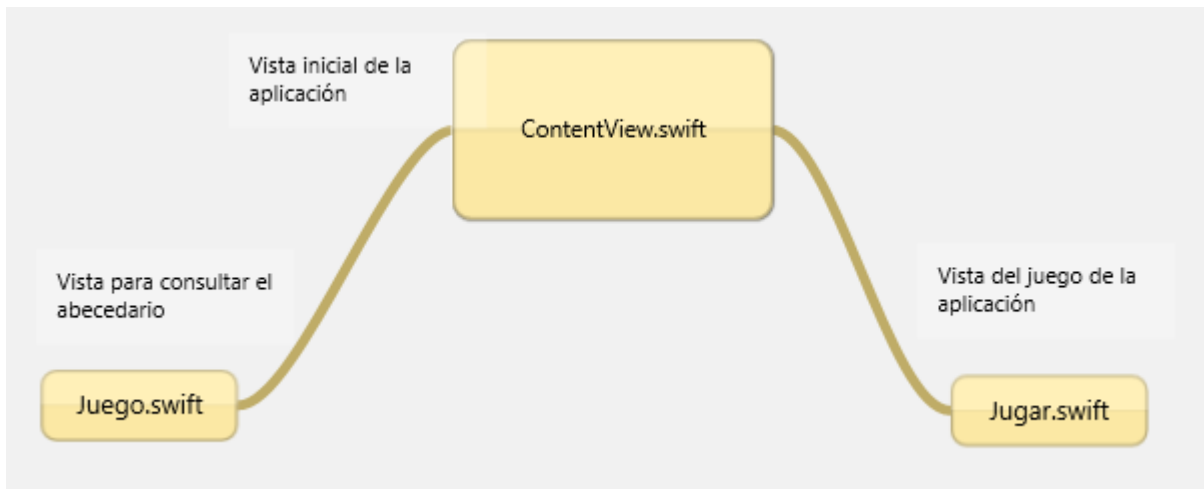
## Descripción del proyecto

signALL es una aplicación escrita en Swift y SwiftUI con el propósito de enseñar el lenguaje de señas, esto con la finalidad de romper la barrera entre las personas oyentes y no oyentes.

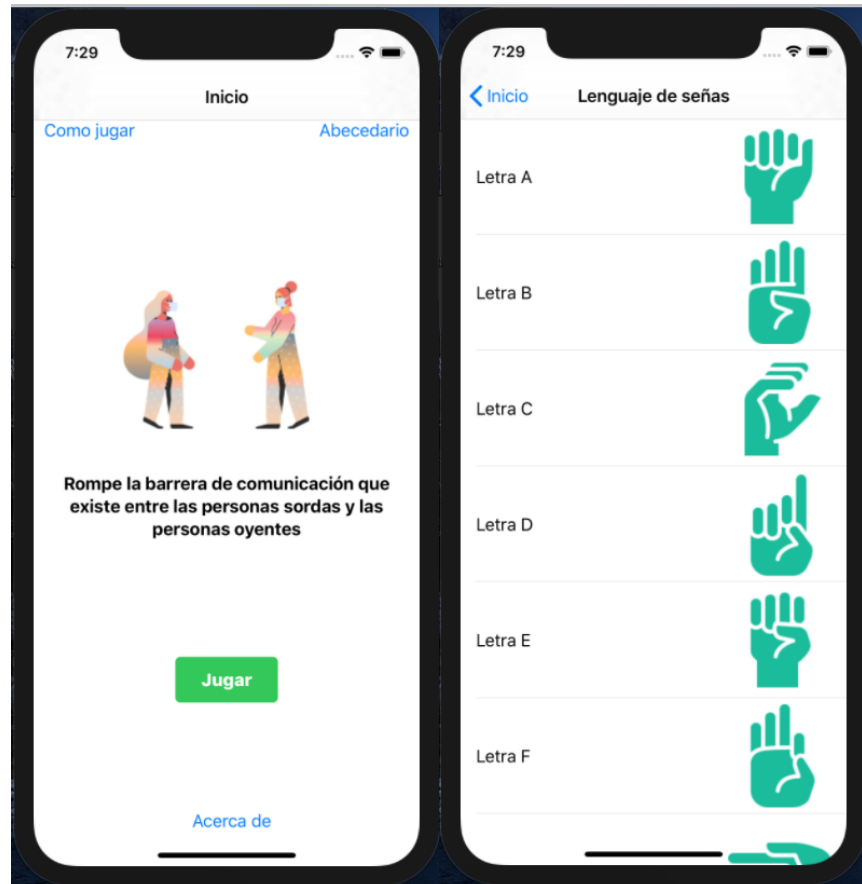
## Diseño de app y navegación

### *Navegación*

La aplicación de signALL cuenta con tres pantallas de navegación:



### *Diseño*



## **Características funcionales**

El usuario dentro de esta aplicación podrá consultar el abecedario (véase vista 2) y podrá acceder al juego (vista 3) que consiste en:

La pantalla mostrará cuatro botones, y una imagen central.

El usuario deberá presionar jugar al entrar a esta pantalla, lo cual activará un Timer el cual brindará 60 segundos para tratar de relacionar la mayor cantidad de letras posibles con la imagen correspondiente.

## Código

### ContentView.swift

```
//
//  ContentView.swift
//  languageGame
//
//  Created by Jair Luna on 14/07/20.
//  Copyright © 2020 Jair Luna. All rights reserved.
//

import SwiftUI

struct ContentView: View {
    @State private var alertaInfo = false
    @State private var alertaAbout = false
    @State private var selection: String? = nil;

    var body: some View {
        NavigationView {
            VStack(alignment: .center){
                HStack{
                    Button(action: {
                        self.alertaInfo = true
                    }) {
                        Text("Como jugar")
                    }
                    .padding(.leading, 10.0)
                    .alert(isPresented: $alertaInfo) { () -> Alert in
                        Alert(title: Text("Como Jugar"), message: Text("
Una imagen será presentada al centro de la pantalla cada turno. Usted de
berá presionar el boton que muestre la opción correcta de letra para obt
ener puntos."), dismissButton: .default(Text("¡Entiendo!")))
                    }
                }
                Spacer()
                NavigationLink(destination: Juego()) {
                    Text("Abecedario")
                }
                .padding(.trailing, 10.0)
            }
            Spacer()
            Image("front")
            Text("Rompe la barrera de comunicación que existe entre
las personas sordas y las personas oyentes")
                .fontWeight(.bold)
                .multilineTextAlignment(.center)
        }
    }
}
```

```

        .lineLimit(3)
        .padding(.all, 10.0)
        Spacer()
        NavigationLink(destination: Jugar(), tag: "Game", selection: $selection) {
            EmptyView()
        }
        Button(action: {
            self.selection = "Game"
        }) {
            Text("Jugar")
                .fontWeight(.heavy)
                .multilineTextAlignment(.center)
                .padding(.horizontal, 25.0)
                .padding(.vertical, 12.0)
                .background(Color.green)
                .foregroundColor(.white)
                .cornerRadius(5)
        }
        Spacer()
        Button(action: {
            self.alertaAbout = true
        }) {
            Text("Acerca de")
        }
        .padding(.leading, 10.0)
        .alert(isPresented: $alertaAbout) { () -> Alert in
            Alert(title: Text("ITL México"), message: Text("App
desarrollada por Arel Jair Monreal Luna en el Instituto Tecnológico de la
Laguna"), dismissButton: .default(Text(";Asombroso!")))
        }
    }
    .navigationBarTitle("Inicio", displayMode: .inline)
}

}

}

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

```



## Juego.swift

```
//
//  Juego.swift
//  languageGame
//
//  Created by Jair Luna on 14/07/20.
//  Copyright © 2020 Jair Luna. All rights reserved.
//

import SwiftUI

struct Letra: Identifiable {
    var id = UUID()
    var name: String
}

let letras: [Letra] = [
    Letra(name: "A"), Letra(name: "B"), Letra(name: "C"),
    Letra(name: "D"), Letra(name: "E"), Letra(name: "F"),
    Letra(name: "G"), Letra(name: "H"), Letra(name: "I"),
    Letra(name: "J"), Letra(name: "K"), Letra(name: "L"),
    Letra(name: "M"), Letra(name: "N"), Letra(name: "O"),
    Letra(name: "P"), Letra(name: "Q"), Letra(name: "R"),
    Letra(name: "S"), Letra(name: "T"), Letra(name: "U"),
    Letra(name: "V"), Letra(name: "W"), Letra(name: "X"),
    Letra(name: "Y"), Letra(name: "Z")
]

struct Juego: View {
    var body: some View {
        List(letras) { letra in
            HStack{
                Text("Letra " + letra.name)
                Spacer()
                Image(letra.name)
            }
        }.navigationBarTitle("Lenguaje de señas", displayMode: .inline
    )

        .navigationBarHidden(false)
    }
}

struct Juego_Previews: PreviewProvider {
    static var previews: some View {
        Juego()
    }
}
```

```
}  
}
```

## Jugar.swift

```
//  
//  Jugar.swift  
//  languageGame  
//  
//  Created by Jair Luna on 14/07/20.  
//  Copyright © 2020 Jair Luna. All rights reserved.  
//  
  
import SwiftUI  
  
struct Jugar: View {  
    let letras: [Letra] = [  
        Letra(name: "A"), Letra(name: "B"), Letra(name: "C"),  
        Letra(name: "D"), Letra(name: "E"), Letra(name: "F"),  
        Letra(name: "G"), Letra(name: "H"), Letra(name: "I"),  
        Letra(name: "J"), Letra(name: "K"), Letra(name: "L"),  
        Letra(name: "M"), Letra(name: "N"), Letra(name: "O"),  
        Letra(name: "P"), Letra(name: "Q"), Letra(name: "R"),  
        Letra(name: "S"), Letra(name: "T"), Letra(name: "U"),  
        Letra(name: "V"), Letra(name: "W"), Letra(name: "X"),  
        Letra(name: "Y"), Letra(name: "Z")  
    ]  
  
    @State var puntuacion: Int = 0  
    @State var aciertos: Int = 0  
    @State var errores: Int = 0  
    @State var letraActual: String = "question"  
    @State var juegoActual: [Letra] = [  
        Letra(name: "?"), Letra(name: "?"), Letra(name: "?")  
    ]  
  
    @State var isCorrect: String = ""  
    @State var tiempo: Int = 60  
    @State var finalizo = false  
  
    @State private var countdownTimer: Timer?  
  
    var body: some View {  
  
        VStack{
```

```

        Spacer()
        Group {
            Button(action: {
                self.newRound()
                self.iniciarJuego()
                if(self.tiempo <= 0) {
                    self.finalizo = true
                }
            }) {
                Text("Jugar")
                .fontWeight(.heavy)
                .multilineTextAlignment(.center)
                .padding(.horizontal, 25.0)
                .padding(.vertical, 12.0)
                .background(Color.green)
                .foregroundColor(.white)
                .cornerRadius(5)
            }.alert(isPresented: self.$finalizo) {
                Alert(title: Text("El juego finalizó"), message: Text(
                    "Su puntuación fue \(${self.puntuacion}"), dismissButton: .default(Text(
                        "¡Ok!")))
            }
            Spacer()
            Text(String(self.tiempo))
                .font(.largeTitle)
                .fontWeight(.bold)
                .foregroundColor(Color.gray)
        }
        Spacer()
        Group {
            Text("Esta es la letra...")
                .font(.title)
                .fontWeight(.bold)
            Image(letraActual)
            Spacer()
            HStack {
                Spacer()
                Button(action: {
                    self.comprobar(letra: self.juegoActual[0].name)
                }) {
                    Text(String(juegoActual[0].name))
                        .fontWeight(.heavy)
                        .multilineTextAlignment(.center)
                        .padding(.horizontal, 25.0)
                        .padding(.vertical, 12.0)
                        .background(Color.green)
                }
            }
        }
    }
}

```

```

        .foregroundColor(.white)
        .cornerRadius(5)
    }
    Spacer()
    Button(action: {
        self.comprobar(letra: self.juegoActual[1].name)
    }) {
        Text(String(juegoActual[1].name))
            .fontWeight(.heavy)
            .multilineTextAlignment(.center)
            .padding(.horizontal, 25.0)
            .padding(.vertical, 12.0)
            .background(Color.green)
            .foregroundColor(.white)
            .cornerRadius(5)
    }
    Spacer()
    Button(action: {
        self.comprobar(letra: self.juegoActual[2].name)
    }) {
        Text(String(juegoActual[2].name))
            .fontWeight(.heavy)
            .multilineTextAlignment(.center)
            .padding(.horizontal, 25.0)
            .padding(.vertical, 12.0)
            .background(Color.green)
            .foregroundColor(.white)
            .cornerRadius(5)
    }
    Spacer()
}
Spacer()
Image(self.isCorrect)
    .frame(width: 48.0, height: 48.0)
Spacer()
HStack{
    Spacer()
    Text("Puntuacion: ")
    Text(String(self.puntuacion))
    Spacer()
    Text("Aciertos: ")
    Text(String(self.aciertos))
    Spacer()
    Text("Errores: ")
    Text(String(self.errores))
}

```

```

        Spacer()
    }
    Spacer()
}

}

func iniciarJuego() {
    self.tiempo = 60
    self.errores = 0
    self.puntuacion = 0
    self.aciertos = 0
    self.countdownTimer = Timer.scheduledTimer(withTimeInterval: 1, r
epeats: true, block: { _ in
        guard self.tiempo > 0 else {
            self.countdownTimer?.invalidate()
            self.finalizo = true
            return
        }
        self.tiempo -= 1
    })
}

func comprobar(letra: String) {
    if letra == self.letraActual {
        puntuacion += 50
        self.aciertos += 1
        self.isCorrect = "ok"
    } else {
        self.errores += 1
        self.isCorrect = "no"
    }
    newRound()
}

func newRound() {
    getRandomLetter()
    self.letraActual = juegoActual[Int.random(in: 0...juegoActual.co
unt-1)].name
}

func getRandomLetter() {
    var letrasB = letras
    for x in 0...2 {
        let indiceLetra = Int.random(in: 0...letrasB.count-1)

```

```

        let letra = letrasB[indiceLetra]
        juegoActual[x] = letra
        letrasB.remove(at: indiceLetra)
    }
}

}

struct Jugar_Previews: PreviewProvider {
    static var previews: some View {
        Jugar()
    }
}

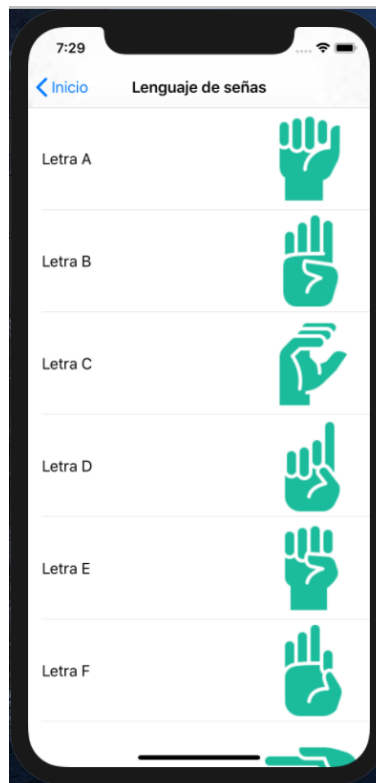
```

## Resultados

### Pantalla inicial



### Pantalla de abecedario



## Pantalla de Juego





## **Anexos**

Repositorio de Github

<https://github.com/m00nreal/signALL>

Video de resultados

<https://www.youtube.com/watch?v=cm8Gyek2SJ4>