

PRÁCTICA 3



Programación en iOS: StoryBoards

Alumno: Elías Jiménez Hernández

Cuenta: 227243-3

Carrera: Ing. Mecatrónica y Sistemas Ciber Físicos

Materia: Aplicaciones Móviles y laboratorio

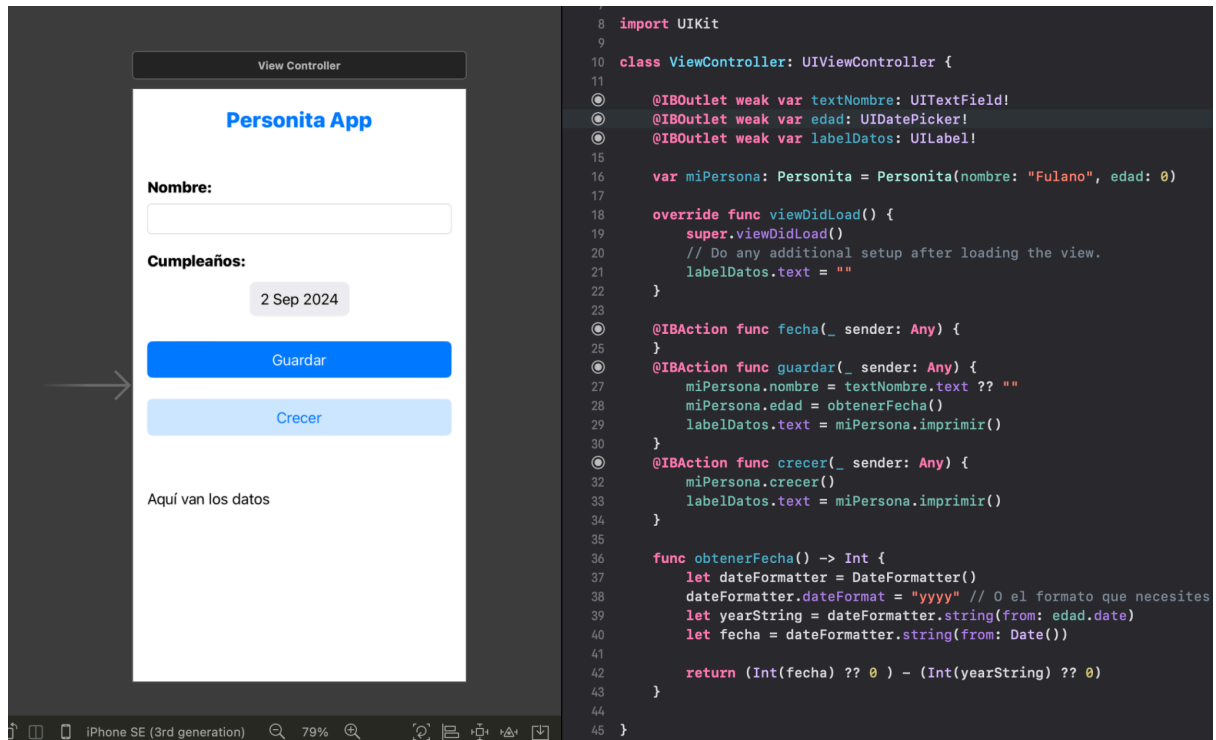
Profesor: Ing. Omar Vázquez González

Descripción

Esta práctica está diseñada para fortalecer tus habilidades en el desarrollo de aplicaciones iOS utilizando Swift u Objective-C. A través de una serie de ejercicios prácticos, aplicarás conceptos fundamentales como Outlets, Actions, y el uso de controles como Date Picker y CustomPicker. Además, aprenderás a manejar imágenes y a importar archivos en tus proyectos.

La práctica incluye la creación de la aplicación "I AM RICH" en iOS, así como el desarrollo de una aplicación de formulario que recoja y almacene información personal, consolidando así tus conocimientos en la gestión de datos y la interacción con el usuario. Esta experiencia te proporcionará una base sólida para enfrentar desafíos más avanzados en el desarrollo de aplicaciones móviles.

- Los ejercicios vistos en clase.



```
import Foundation

class Personita {
    var nombre: String
    var edad: Int

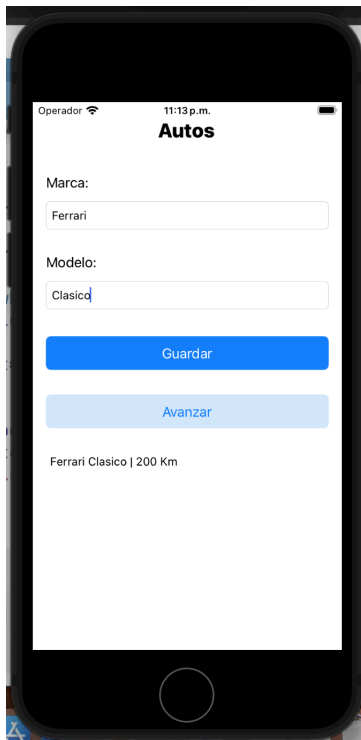
    init(nombre: String, edad: Int) {
        self.nombre = nombre
        self.edad = edad
    }

    func crecer() {
        edad += 1
    }

    func crecer(x: Int) {
        edad += x
    }

    func imprimir()-> String{
        return "Hola soy \(self.nombre) y tengo \(self.edad) años"
    }
}
```

- Ejercicio Auto en iOS (Ver el video)



```
import Foundation

class Auto {
    var marca: String
    var modelo: String
    var posicion: Int

    init() {
        marca = ""
        modelo = ""
        posicion = 0
    }

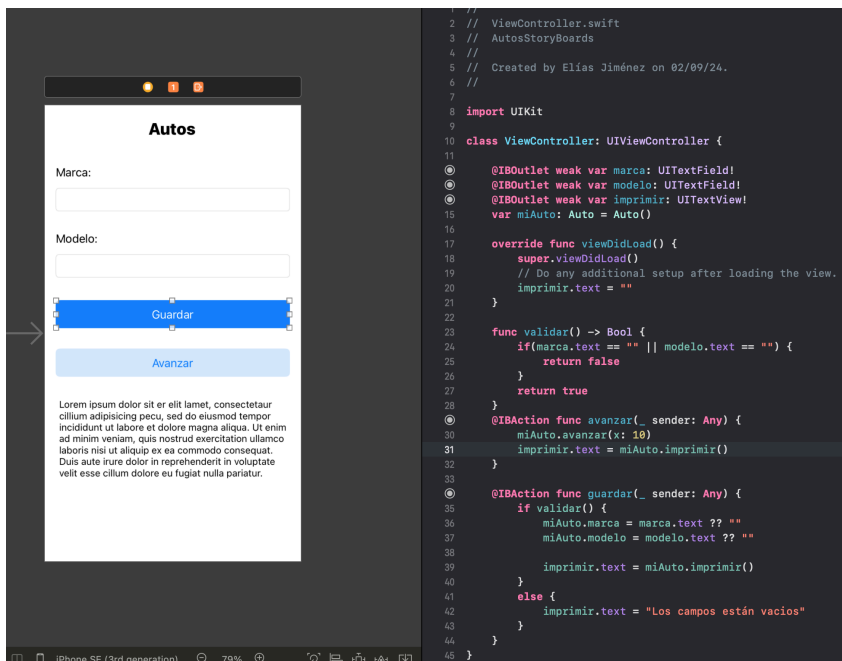
    init(marca:String, modelo:String) {
        self.modelo = modelo
        self.marca = marca
        self.posicion = 0
    }

    init(marca:String, modelo:String, posicion:Int) {
        self.modelo = modelo
        self.marca = marca
        self.posicion = posicion
    }

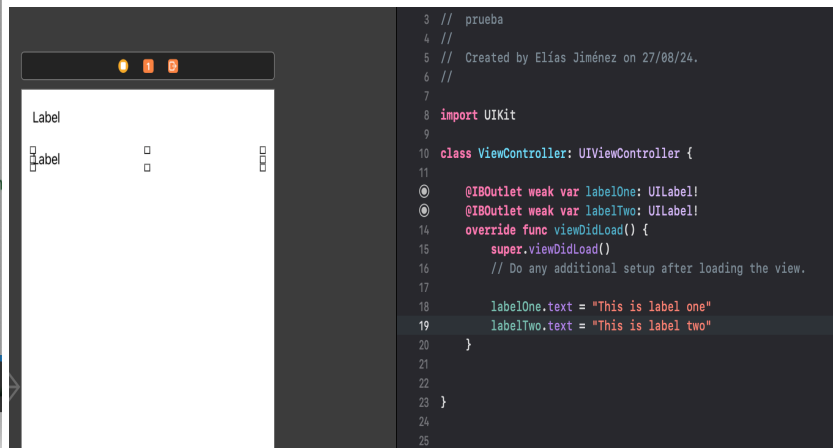
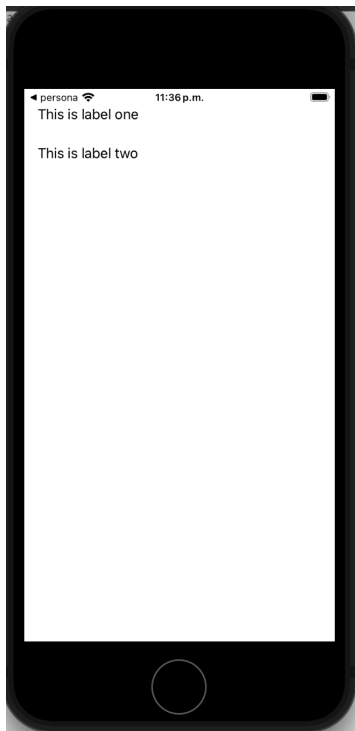
    func avanzar() {
        posicion += 1
    }

    func avanzar(x: Int) {
        posicion += x
    }

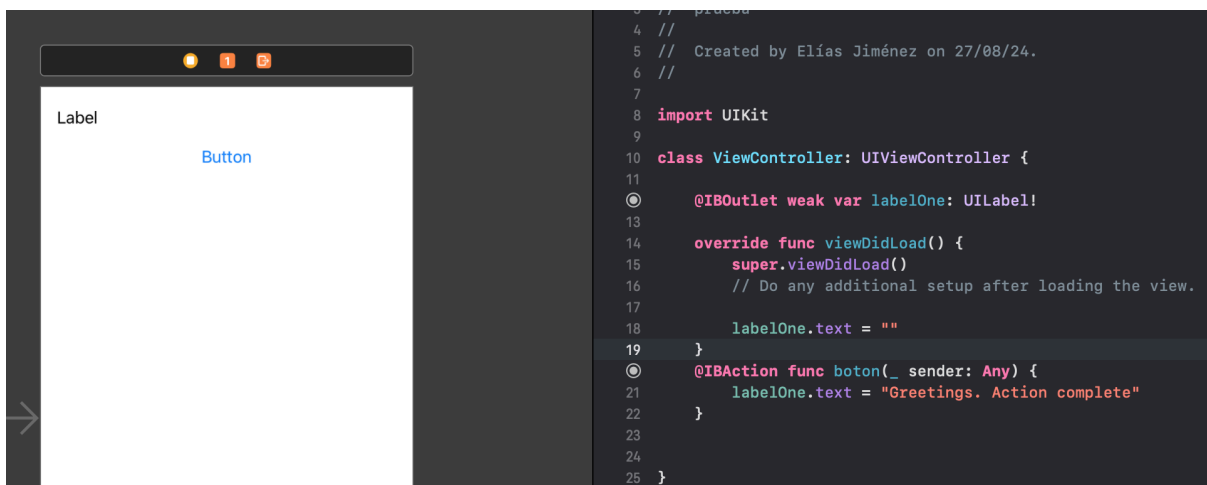
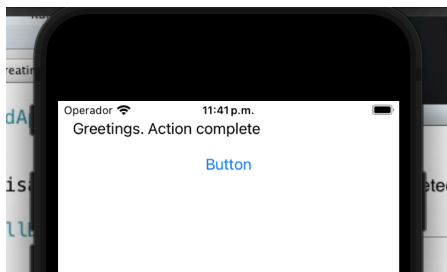
    func imprimir() -> String {
        return "\(marca) \(modelo) | \(posicion) Km"
    }
}
```



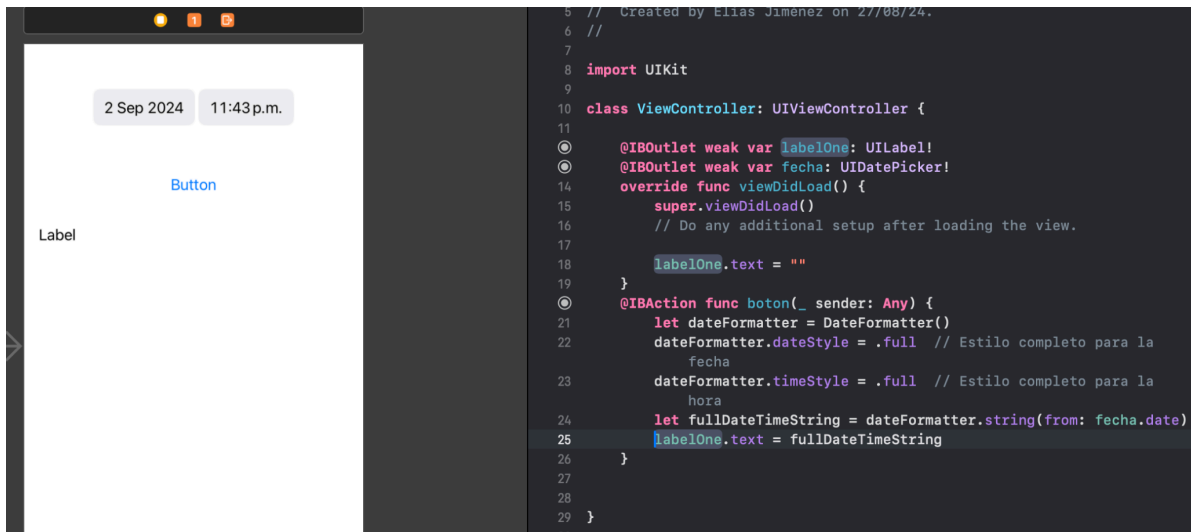
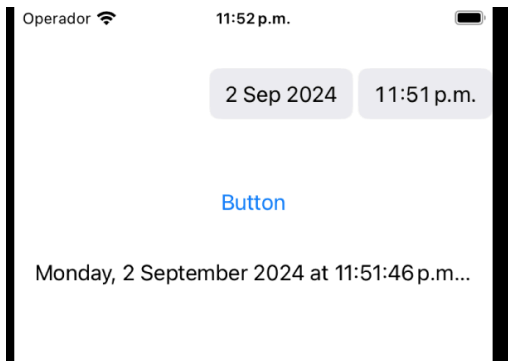
- El contenido de los siguientes videos del curso iOS Básico:
 - I04. Outlets



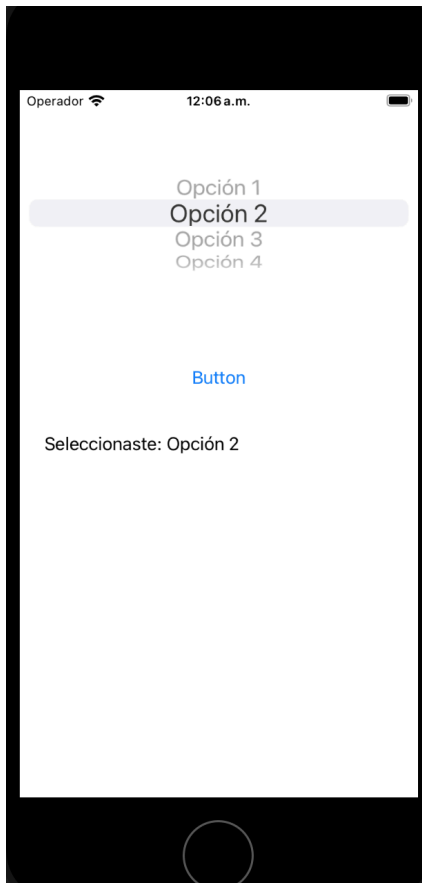
- I05. Actions



○ I13. Date Picker



○ I14. CustomPicker



```
import UIKit
```

```
class ViewController: UIViewController, UIPickerViewDelegate,  
UIPickerViewDataSource {
```

```
    @IBOutlet weak var picker: UIPickerView!
```

```
    @IBOutlet weak var labelOne: UILabel!
```

```
    // Array de valores que se mostrarán en el picker
```

```
    let pickerData = ["Opción 1", "Opción 2", "Opción 3", "Opción 4"]
```

```
override func viewDidLoad() {  
    super.viewDidLoad()  
  
    // Configurar el delegado y fuente de datos del picker  
    picker.delegate = self  
    picker.dataSource = self  
  
    labelOne.text = ""  
}  
  
// MARK: - UIPickerViewDataSource  
  
func numberOfComponents(in pickerView: UIPickerView) -> Int {  
    return 1 // Solo 1 columna  
}  
  
func pickerView(_ pickerView: UIPickerView,  
numberOfRowsInComponent component: Int) -> Int {  
    return pickerData.count // Número de filas igual al número de  
    elementos en pickerData  
}
```



```
// MARK: - UIPickerViewDelegate
```

```
func pickerView(_ pickerView: UIPickerView, titleForRow row: Int,  
forComponent component: Int) -> String? {
```

```
    return pickerData[row] // Retorna el título para cada fila  
}
```

```
@IBAction func boton(_ sender: Any) {
```

```
    // Obtener el índice seleccionado
```

```
    let selectedRow = picker.selectedRow(inComponent: 0)
```

```
    // Obtener el valor correspondiente al índice
```

```
    let selectedValue = pickerData[selectedRow]
```

```
    // Mostrar el valor seleccionado en el label
```

```
    labelOne.text = "Seleccionaste: \(selectedValue)"
```

```
}
```

```
}
```

- El contenido de los siguientes videos del curso iOS Intermedio:

- IA19. Images

@IBOutlet weak var imagen: UIImageView!

override func viewDidLoad() {

super.viewDidLoad()

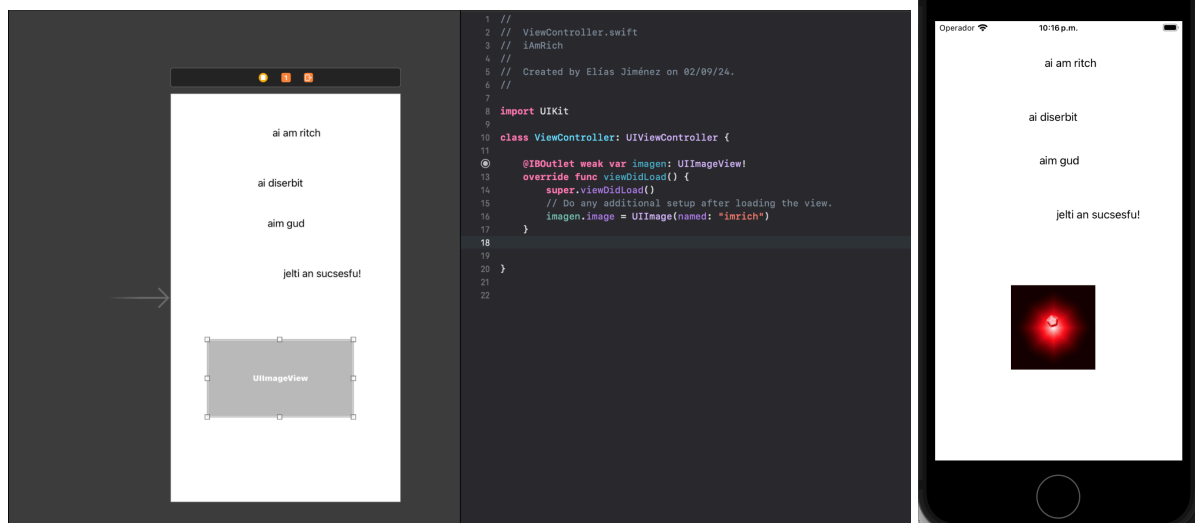
// Do any additional setup after loading the view.

imagen.image = UIImage(named: "imrich")

}

- IA17. Importing Files

- Crear la aplicación: "I AM RICH" en iOS



- Ejercicio de Persona (Una App con un formulario donde acepte sus datos personales y los almacene en una clase y los muestre). Puede ser en Objective C o Swift:
 - Nombre
 - Apellido Paterno
 - Apellido Materno
 - e-mail

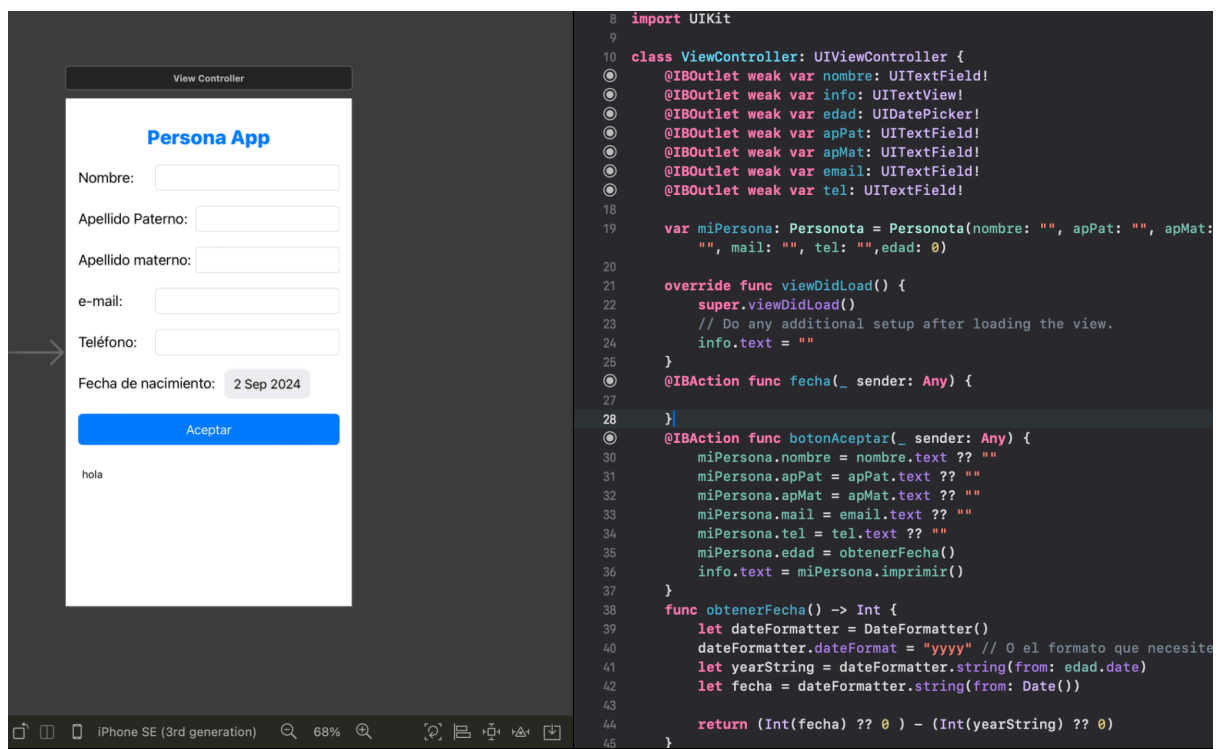
- telefono
- fecha de nacimiento

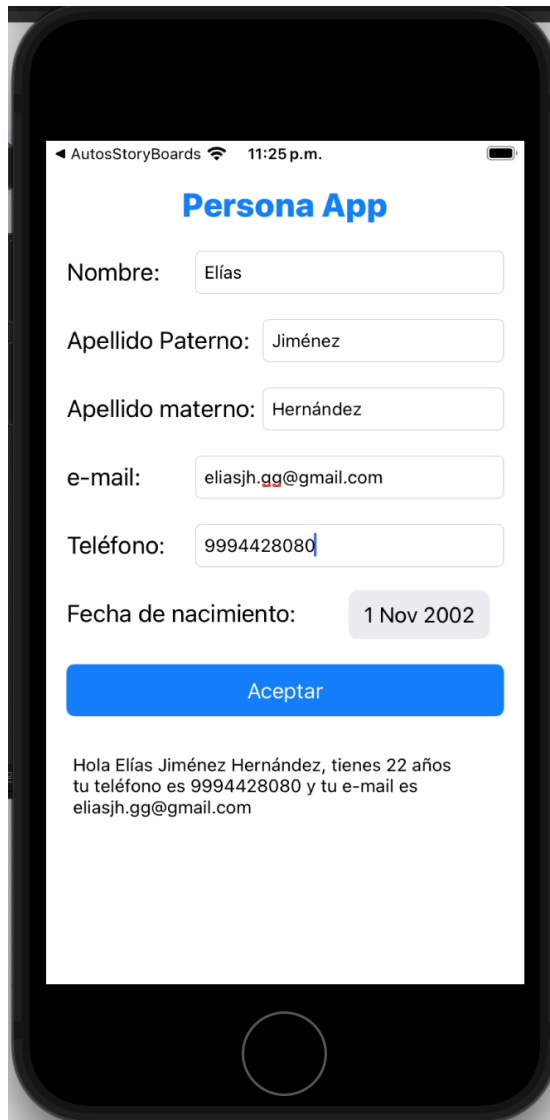
import Foundation

```
class Personota {
    var nombre: String
    var apPat: String
    var apMat: String
    var mail: String
    var tel: String
    var edad: Int

    init(nombre: String, apPat: String, apMat: String, mail: String, tel: String, edad: Int) {
        self.nombre = nombre
        self.apMat = apMat
        self.apPat = apPat
        self.mail = mail
        self.tel = tel
        self.edad = edad
    }

    func imprimir()-> String{
        return "Hola \(self.nombre) \(self.apPat) \(self.apMat), tienes \(self.edad) años\ntu
        teléfono es \(self.tel) y tu e-mail es \(self.mail)"
    }
}
```





Conclusión:

La práctica ha permitido una inmersión profunda en la programación de iOS, proporcionando herramientas y conocimientos necesarios para desarrollar aplicaciones funcionales y bien estructuradas. Los ejercicios propuestos no solo han permitido aplicar conceptos teóricos en situaciones prácticas, sino que también han fomentado el desarrollo de habilidades cruciales en la creación y gestión de interfaces, el manejo de datos y la integración de recursos multimedia. Con estas habilidades, los desarrolladores están mejor equipados para enfrentar desafíos más avanzados y construir aplicaciones móviles completas y efectivas.