



Universidad Nacional Autónoma de México

Matemáticas Aplicadas y Computación

Temas Selectos de Computacion

Práctica “Adivina Quien”

Profesor
Eduardo Eloy Loza Pacheco

Alumno
Hernández Espinoza Gabriel Emanuel

Indice

Objetivo.....	1
Materiales y Metodos.....	1
Resumen.....	1
Antecedentes Teoricos.....	2
Desarollo.....	3
Conclusiones.....	5
Referencias.....	6

Objetivo

Desarrollar una aplicación interactiva en iOS mediante Xcode que permita realizar un test de personalidad basado en la serie Pokémon, utilizando controles deslizantes (UISlider), condicionales y segues para mostrar diferentes resultados según las respuestas del usuario.

Materiales y Métodos

Para la elaboración de esta práctica se utilizó una computadora Mac con sistema operativo macOS y el entorno de desarrollo integrado Xcode. Se creó un proyecto de tipo 'App' para iPhone, empleando los componentes UISlider, UILabel y UIButton para la captura e interpretación de datos del usuario. El código fue desarrollado en el lenguaje Swift, haciendo uso de estructuras condicionales y funciones para analizar los valores ingresados en los sliders. Además, se implementaron segues tipo 'Present Modally' para redirigir al usuario a diferentes pantallas según el resultado obtenido en el test. Finalmente, la aplicación fue probada en el simulador de iPhone, verificando la correcta vinculación entre vistas y la precisión de la lógica condicional.

Resumen

En esta práctica se desarrolló una aplicación móvil denominada 'Adivina quién', la cual funciona como un test de personalidad inspirado en Pokémon. El objetivo principal fue comprender la implementación de controles deslizantes (UISlider) y la navegación entre vistas utilizando segues en el entorno Xcode. La aplicación permite al usuario ajustar tres variables —tímido, curioso y seguro— mediante sliders, cuyos valores determinan un resultado final: Pichu, Pikachu o Raichu.

El desarrollo del programa involucró el uso de estructuras condicionales para comparar los valores ingresados y definir cuál característica domina. Posteriormente, se ejecuta una transición entre vistas utilizando segues, mostrando la imagen o pantalla correspondiente al Pokémon asignado. Además, se implementaron etiquetas dinámicas que actualizan los valores de los sliders en tiempo real, reforzando la interacción y la retroalimentación inmediata.

Esta práctica permitió integrar el manejo de controles de interfaz, la lógica de decisión y la navegación entre vistas, conceptos fundamentales en el desarrollo de aplicaciones iOS. El resultado fue una aplicación funcional, didáctica y visualmente atractiva, que combina elementos de programación estructurada con diseño de experiencia de usuario.

Antecedentes Teóricos

El diseño de aplicaciones interactivas en iOS implica la integración de diversos componentes de interfaz y estructuras lógicas que permitan interpretar la entrada del usuario. En este contexto, los controles UISlider son herramientas que facilitan la captura de valores continuos dentro de un rango, siendo ampliamente utilizados en aplicaciones de evaluación o ajuste de parámetros. Estos controles envían eventos al sistema que pueden ser gestionados mediante acciones (@IBAction), lo cual permite actualizar dinámicamente otros elementos visuales como etiquetas (UILabel) [1].

El uso de condicionales (if-else) constituye la base de la lógica de decisión en la programación. En Swift, estas estructuras permiten comparar valores y ejecutar bloques de código específicos según las condiciones cumplidas. Esta funcionalidad es esencial para determinar resultados en aplicaciones tipo test o juegos, donde se requiere evaluar múltiples posibilidades [2].

Los segues son mecanismos nativos de Xcode que permiten realizar transiciones entre vistas dentro del patrón de diseño Modelo-Vista-Controlador (MVC). Existen diferentes tipos de segues, como 'Show', 'Push' y 'Present Modally'. En este proyecto se utilizó el último, que permite presentar una nueva vista sobre la actual sin necesidad de una barra de navegación. El uso adecuado de segues facilita la modularidad y mejora la experiencia del usuario [3].

Finalmente, desde el punto de vista del diseño, las Human Interface Guidelines (HIG) de Apple recomiendan que las aplicaciones proporcionen retroalimentación inmediata y una navegación intuitiva. El proyecto 'Adivina quién' sigue estos principios al actualizar los valores de los sliders en tiempo real y ofrecer transiciones visuales suaves entre vistas, mejorando la percepción del usuario [4].

Desarrollo

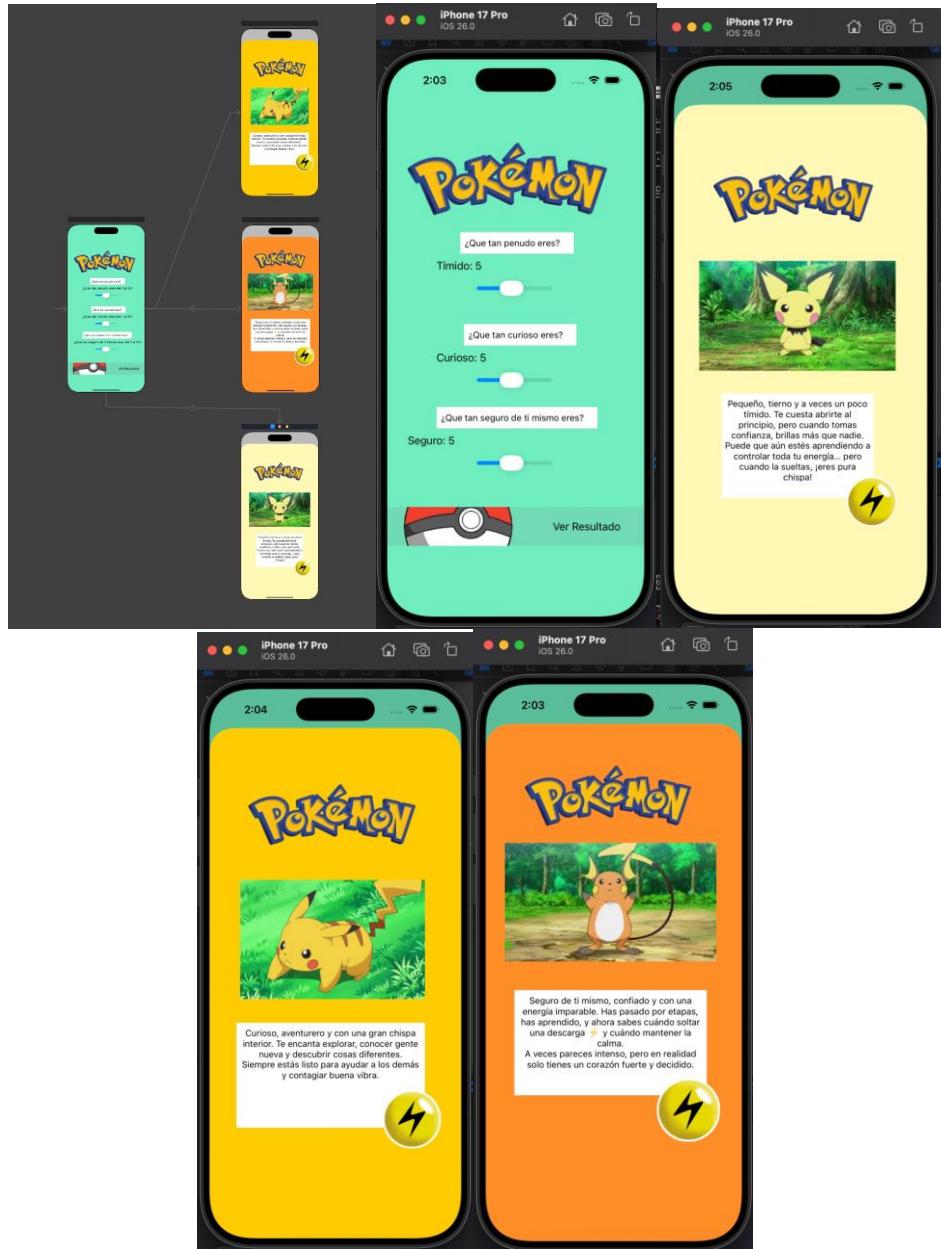
La creación de la aplicación comenzó con la configuración de un nuevo proyecto en Xcode utilizando la plantilla 'App' para iPhone. Dentro del storyboard se diseñó la interfaz principal, incorporando tres controles deslizantes (UISlider) para capturar los niveles de timidez, curiosidad y seguridad del usuario. Junto a estos, se añadieron etiquetas (UILabel) que mostraban los valores actuales de cada slider, actualizándose de forma dinámica conforme el usuario los modificaba.

En el código, se definieron conexiones mediante @IBOutlet para enlazar los componentes visuales con las variables del controlador. El método viewDidLoad() inicializó los valores predeterminados de los sliders en 5 y llamó a la función updateLabels() para actualizar las etiquetas al inicio de la aplicación. Posteriormente, se creó una función sliderChanged() vinculada a la acción @IBAction que se ejecuta cada vez que el usuario mueve un control. Esta función redondea los valores de los sliders y actualiza los textos mostrados, garantizando coherencia entre la entrada y la visualización.

El corazón lógico de la aplicación se encuentra en la función showResult(), que se ejecuta al presionar el botón 'Ver resultado'. Esta función obtiene los valores de los tres sliders y los compara para determinar cuál rasgo domina. Si el valor de timidez es mayor, el resultado asignado es 'Pichu'; si el valor de curiosidad predomina, se muestra 'Pikachu'; y si la seguridad es superior, el resultado corresponde a 'Raichu'. Con base en esta decisión, se ejecuta el segue correspondiente ('showPichu', 'showPikachu' o 'showRaichu'), utilizando el método performSegue().

Cada segue está configurado en el storyboard con el tipo 'Present Modally', lo que permite una transición visual suave y un flujo de navegación coherente. Este enfoque modular facilita el mantenimiento del proyecto y la ampliación futura a nuevos resultados o categorías.

Finalmente, la aplicación fue probada en el simulador de iPhone, confirmando que los sliders respondían correctamente y que las transiciones entre vistas se realizaban sin errores. El resultado fue una aplicación funcional y entretenida que demuestra la integración de componentes de interfaz, lógica condicional y navegación entre vistas en Xcode.



Conclusiones

Esta práctica permitió afianzar los conocimientos sobre la manipulación de controles de interfaz en iOS, en particular los controles deslizantes (UISlider) y la implementación de segues para la navegación entre vistas. Mediante el desarrollo del test de personalidad 'Adivina quién', se logró aplicar de forma práctica conceptos de programación condicional, vinculación de elementos visuales y diseño de flujos interactivos.

El proyecto reforzó la comprensión del patrón de diseño Modelo-Vista-Controlador (MVC), donde la vista representa los controles y etiquetas, el modelo se asocia a los valores de los sliders, y el controlador gestiona la lógica de decisión y las transiciones. La implementación de condicionales para comparar los valores ingresados permitió crear una experiencia personalizada para cada usuario.

Asimismo, el uso de segues tipo 'Present Modally' demostró la importancia de la navegación estructurada y la interacción visual en el desarrollo móvil. El proyecto cumplió con los principios de las Human Interface Guidelines, proporcionando retroalimentación inmediata y claridad en los resultados mostrados.

En conclusión, esta práctica representó un ejercicio integral de lógica, diseño e interacción en Xcode, consolidando habilidades clave para el desarrollo de aplicaciones móviles dinámicas e interactivas.

Referencias

- [1] Apple, *UIKit Framework Reference*. Apple Developer, 2023. [En línea]. Disponible en: <https://developer.apple.com/documentation/uikit>
- [2] Apple, *Swift Programming Language Guide*. Apple Developer Documentation, 2023. [En línea]. Disponible en: <https://docs.swift.org/swift-book/>
- [3] Apple, *Xcode Overview*. Apple Developer, 2023. [En línea]. Disponible en: <https://developer.apple.com/xcode/>
- [4] Apple, *Human Interface Guidelines*. Apple Developer, 2022. [En línea]. Disponible en: <https://developer.apple.com/design/human-interface-guidelines/>