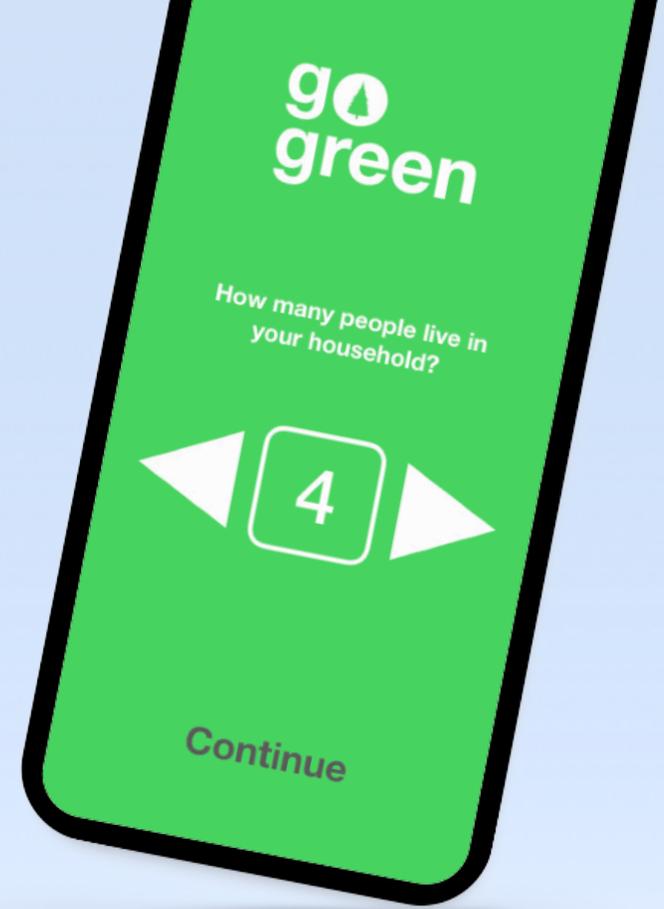
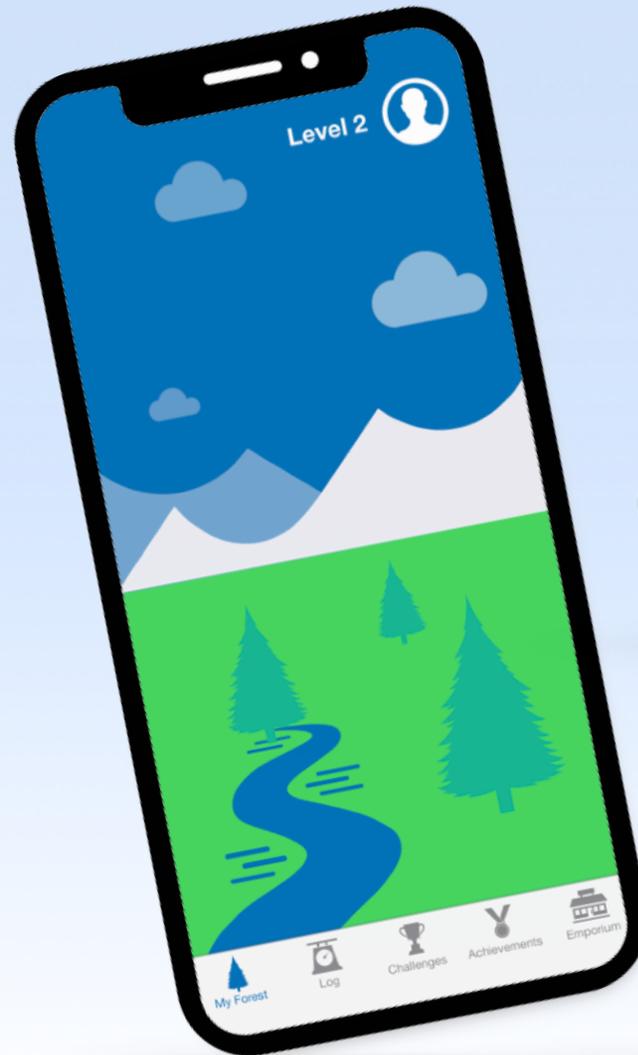




Swift Coding Club



# Xcode

Proyectos de desarrollo de apps

## Proyectos

<b>1</b>	SinglePhoto	3
<b>2</b>	QuestionBot	4
<b>3</b>	ChatBot	5
<b>4</b>	ColorMix	6
<b>5</b>	ElementsQuiz	7
<b>6</b>	AnimalSounds	8
<b>7</b>	RockPaperScissors	9
<b>8</b>	MemeMaker	10

## ¡Te damos la bienvenida al Swift Coding Club!

Al unirse a este club de programación, ya has dado el primer paso para crear tus primeras apps. No te preocupes si llevas muy poco tiempo aprendiendo a programar. Estos proyectos te mostrarán los aspectos básicos tanto de los conceptos de programación como del desarrollo de apps. Cada proyecto toma el anterior como punto de partida y se centra en diferentes aspectos de las apps, de modo que al final tendrás los conocimientos necesarios para crear tu propia app.

Necesitarás el curso [Intro to App Development with Swift \(Introducción al desarrollo de apps con Swift\)](#) para hacer estos proyectos. Asegúrate de cumplir los requisitos previos de cada uno.



# SinglePhoto

1

**Requisitos previos:** Ninguno

**Duración:** 1,5 horas

**Accede:** [Lesson 5](#): First App

---

Descubre lo fácil que es crear tu primera app. Con SinglePhoto, crearás una app que, además de mostrar cosas, como una foto, también reacciona cuando el usuario toca la pantalla. Esta sencilla interacción con la app es la base de muchas apps, así que te vendrá muy bien para ampliar tus habilidades de programación y desarrollo de apps. Este proyecto te permitirá familiarizarte con Xcode, Interface Builder y Simulator, y además aprenderás a combinarlos para crear tus propias apps.



# QuestionBot

## 2

### Requisitos previos:

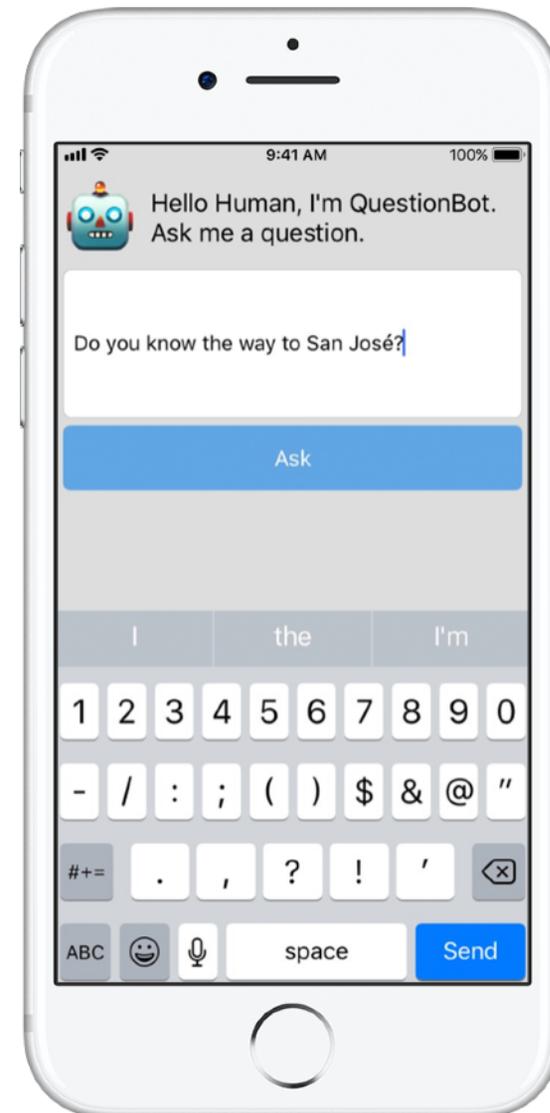
- App SinglePhoto
- Lesson 2: Naming and Identifiers
- Lesson 3: Strings
- Lesson 6: Functions
- Lesson 10: Parameters and Results
- Lesson 11: Making Decisions

**Duración:** 6,5 horas

**Accede:** [Lesson 13: QuestionBot](#)

---

¿Alguna vez has usado una app de juegos de preguntas o has querido saber cómo funciona Siri? Hay muchas apps que parece que tienen un «cerebro» propio. Con QuestionBot, vas a crear una app que contiene el cerebro de un robot que responde de forma diferente en función de las preguntas. Para ello, vas a aprender muchas cosas, como almacenar valores en constantes, representar texto en código y definir entradas y salidas. Este proyecto te ayuda a centrarte en comprender cómo funciona la app y la lógica que se necesita para que parezca que la app tiene un «cerebro».



## Requisitos previos:

- App QuestionBot
- Lesson 14: Arrays and Loops
- Lesson 15: Defining Structures

**Duración:** 3,5 horas

**Accede:** [Lesson 16](#): QuestionBot 2

---

Ahora vamos a hacer que QuestionBot sea aún más inteligente. Quieres usar tu app muchas veces, así que estaría bien que fuese capaz de «recordar» las interacciones pasadas. Con ChatBot, vas a crear una versión mejorada de QuestionBot que conserva un historial de los mensajes intercambiados entre el usuario y el robot. Como ChatBot ya está creado a medias, puedes concentrarte en aprender las habilidades necesarias para crear la parte de la app que lleva el registro de la conversación.



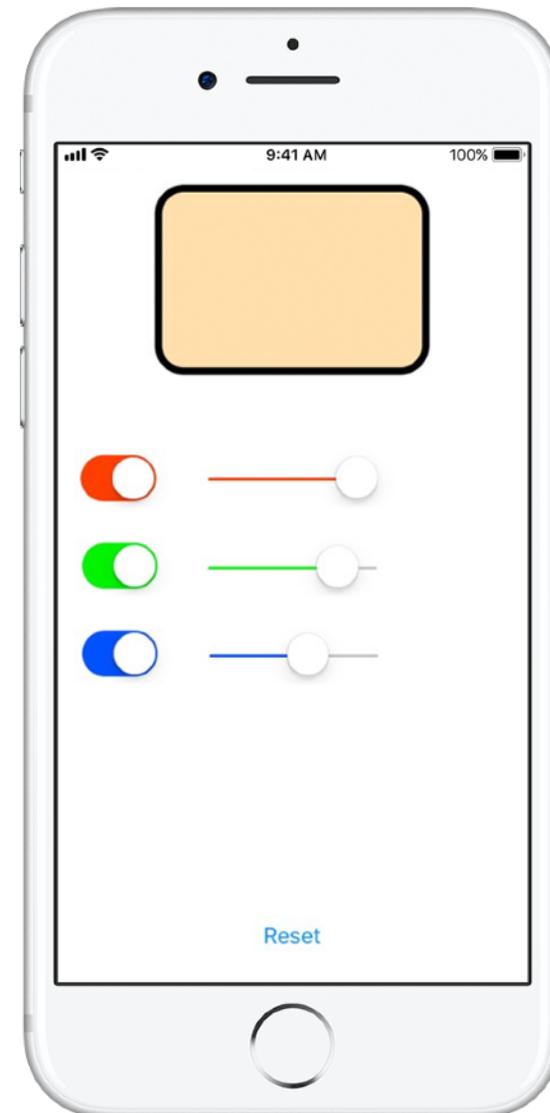
**Requisitos previos:** App ChatBot

**Duración:** 1,5 horas

**Accede:** [Lesson 17](#): Actions and Outlets

---

Piensa en la interfaz de usuario (IU). Hasta ahora, has hecho apps en las que el usuario puede tocar botones e introducir texto. Con ColorMix, también aprenderás a añadir interruptores y reguladores a la IU. Y lo que es más importante, aprenderás a conectar esos elementos visuales de la IU con código de Swift para que funcionen adecuadamente. Al final, tendrás ColorMix, una app que genera colores mezclando rojo, verde y azul.



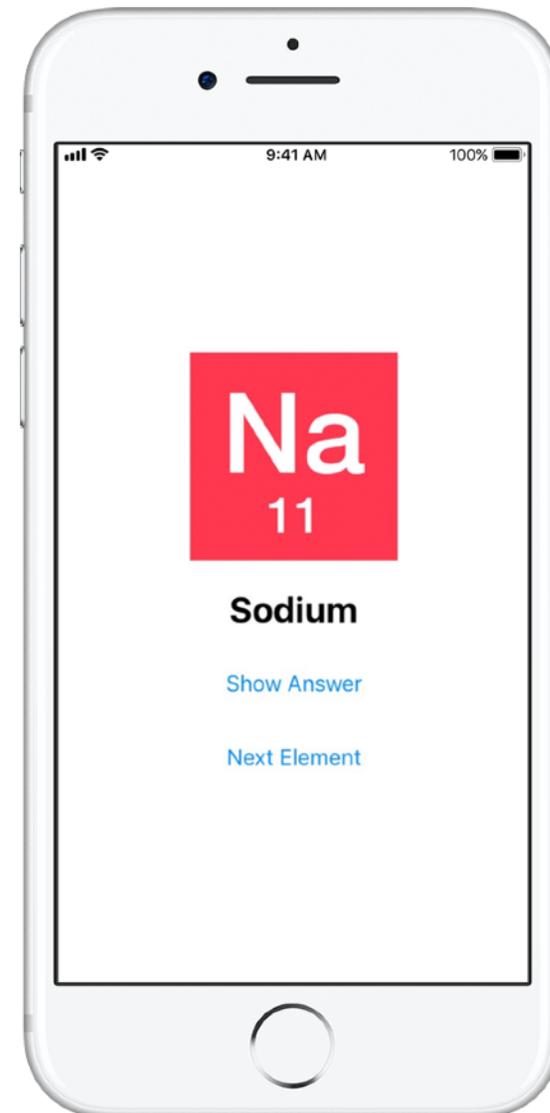
**Requisitos previos:** App ColorMix

**Duración:** 2 horas

**Accede:** [Lesson 18](#): Adaptive User Interfaces

---

La mayoría de la gente descarga apps para resolver un problema en particular, como organizarse, calcular sus finanzas u obtener indicaciones. En ElementsQuiz, vas a crear una app que ayuda a los estudiantes a memorizar los elementos de la tabla periódica. Vas a usar vistas apiladas para colocar fácilmente varias vistas en la pantalla a la vez, y las posiciones se ajustarán para encajar en una amplia gama de dimensiones de pantalla.



# AnimalSounds

6

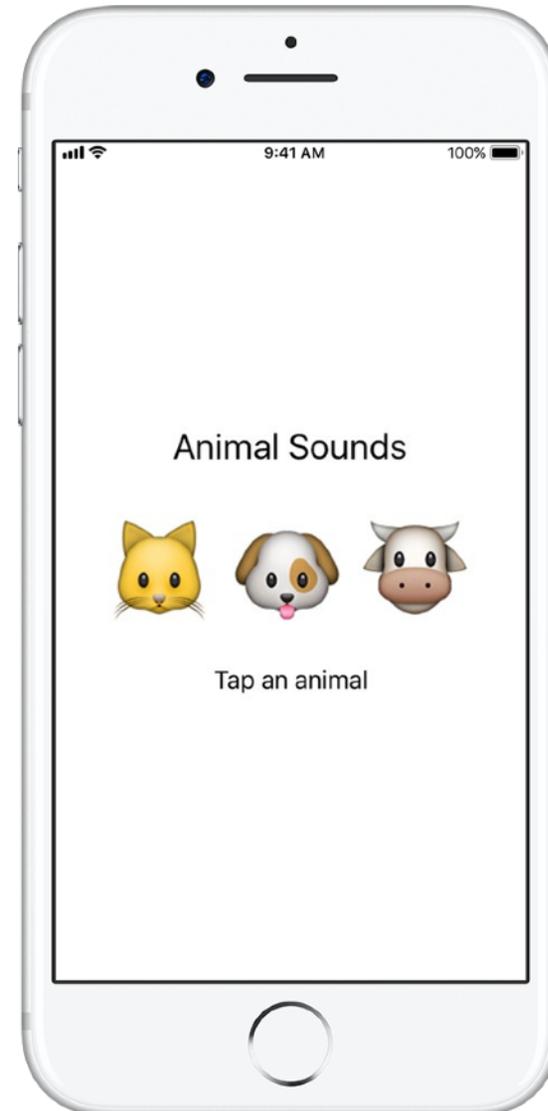
**Requisitos previos:** App ElementsQuiz

**Duración:** 1 hora

**Accede:** [Lesson 18](#): Adaptive User Interfaces

---

A medida que vayas haciendo apps más complejas, es posible que te preguntes si vas a tener que escribir tú todo el código. ¿No lo habrá hecho alguien ya? En AnimalSounds, vas a usar código ya creado que hace todo el trabajo duro necesario para reproducir audio, y también vas a aprender a integrarlo en tu app. A continuación, usarás vistas apiladas dentro de otras vistas apiladas para crear tu interfaz de usuario más avanzada hasta la fecha.



# RockPaperScissors

7

## Requisitos previos:

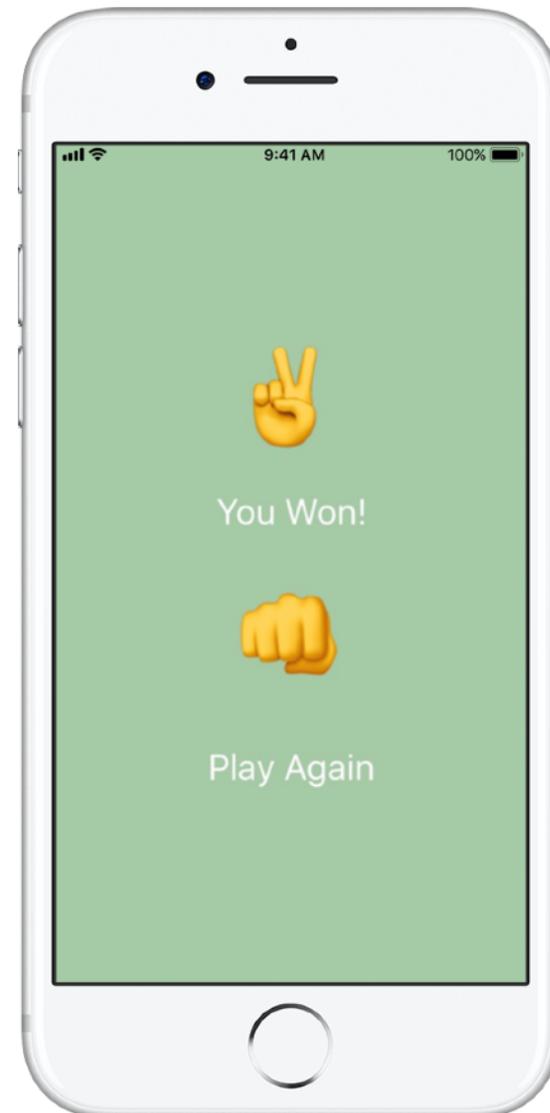
- App AnimalSounds
- Lesson 19: Enumerations

**Duración:** 1,5 horas

**Accede:** [Lesson 20](#): Final Project

---

A menudo tendrás que presentar al usuario una lista de opciones, como una indicación para viajar o un tipo de cocina. En este proyecto, vas a aprender a crear las tres opciones únicas del juego «Piedra, papel o tijera» de modo que el usuario pueda jugar sin parar contra el ordenador. Además, aprenderás a generar números aleatorios para que el ordenador elija una opción al azar.



# MemeMaker

## 8

### Requisitos previos:

- App AnimalSounds
- Lesson 19: Enumerations

**Duración:** 1,5 horas

**Accede:** [Lesson 20](#): Final Project

---

Un meme de internet suele ser una imagen fija con texto arriba y abajo. En MemeMaker vas a aprender a usar controles segmentados para mostrar diferentes textos encima y debajo de una imagen. Como los controles son independientes, puedes combinar el texto para crear combinaciones personalizadas.

