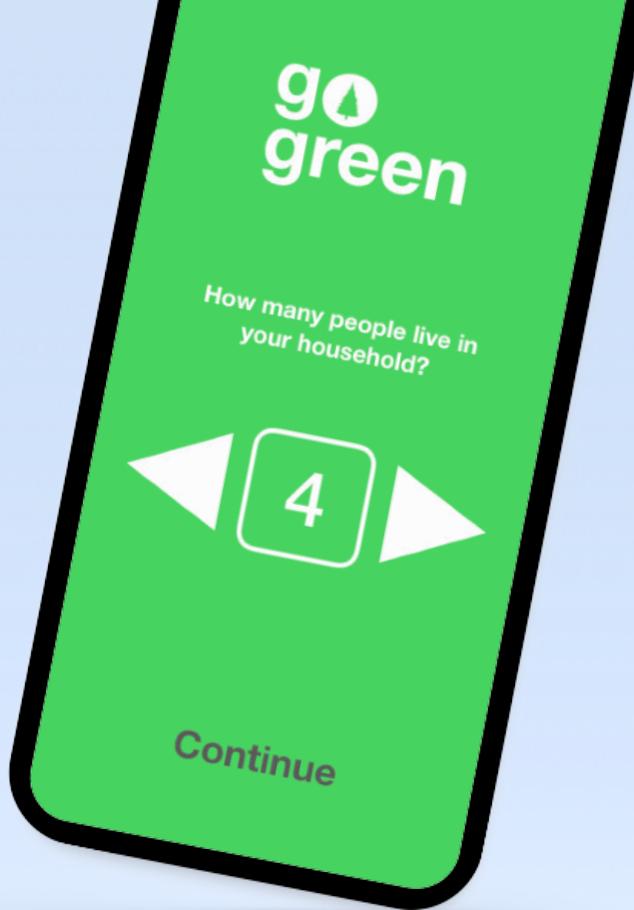
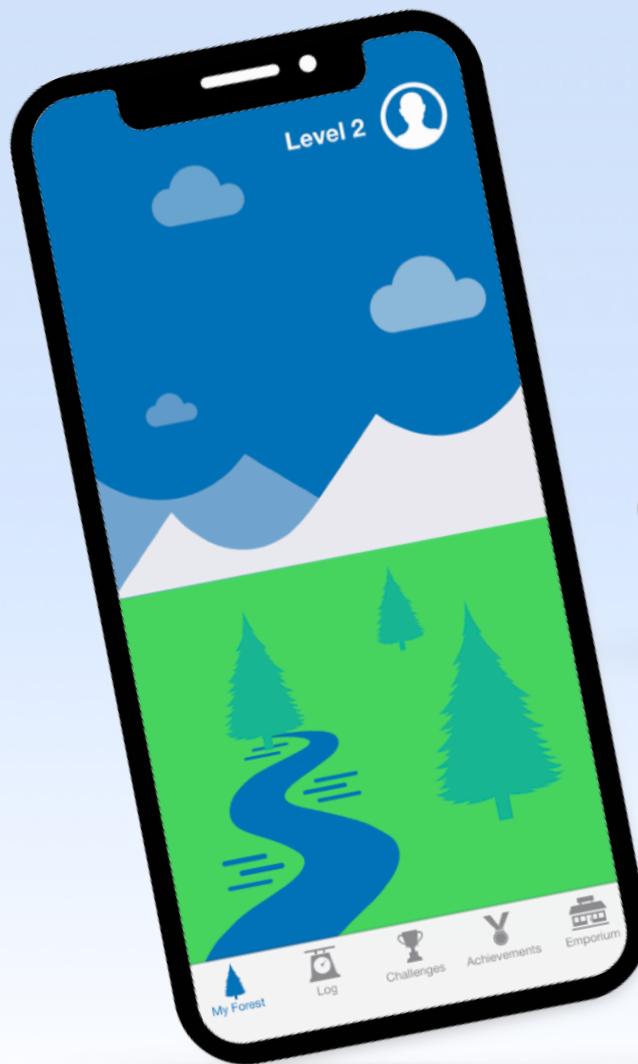




Swift Coding Club



Kit de Xcode

¡Te damos la bienvenida al Swift Coding Club!

Aprender a programar te enseña a resolver problemas y a trabajar en equipo de formas creativas. Y te ayuda a diseñar apps que dan forma a tus ideas.

Los Swift Coding Clubs son una forma divertida de aprender a programar y diseñar apps. Con las actividades basadas en Swift, el lenguaje de programación de Apple, trabajas en equipo mientras aprendes a programar, diseñas prototipos de apps y reflexionas sobre cómo cambiar el mundo que te rodea escribiendo código.

No hace falta que seas profesor o experto en programación para tener un club de programación con Swift. Puedes trabajar los materiales a tu ritmo, lo que te permite aprender junto con el resto de integrantes del club. Y podéis celebrar vuestras ideas y diseños organizando un evento social para presentar las apps que habéis creado.

El kit se divide en tres secciones:



Empieza

Todo lo que necesitas para montar un Swift Coding Club.



Aprende y diseña

Consejos y actividades para diseñar sesiones para el club.



Celebra

Recursos útiles para planificar y organizar un evento social de presentación de apps.

Swift Coding Clubs

Programación por bloques | Entre 8 y 11 años

Aprende los conceptos básicos con apps visuales en el iPad.



Swift Playgrounds | A partir de 11 años

Usa código Swift para aprender los fundamentos de la programación con Swift Playgrounds en el iPad.



Xcode | A partir de 14 años

Aprende a desarrollar apps con Xcode en el Mac.



Empieza



1. Descarga los materiales del club.

Usa AirDrop para compartir estas dos guías con los miembros del club en vuestra primera reunión. También se incluyen en este documento.

Xcode
App Development Projects

Proyectos de desarrollo de apps

Los proyectos de desarrollo guiado de apps presentan las mismas herramientas, técnicas y conceptos que utilizan los profesionales. Aprende conceptos de programación mientras escribes en los playgrounds de Xcode en el Mac.

[Descargar proyectos de desarrollo de apps de Xcode >](#)

Xcode
App Design Journal

Diario de diseño de apps

Explora el proceso de diseño de apps con este diario de Keynote. Haz una lluvia de ideas, planifica, diseña un prototipo y evalúa las ideas para apps de tu club.

[Descargar el diario de diseño de apps de Xcode >](#)



2. Recopila la tecnología que necesitaréis.

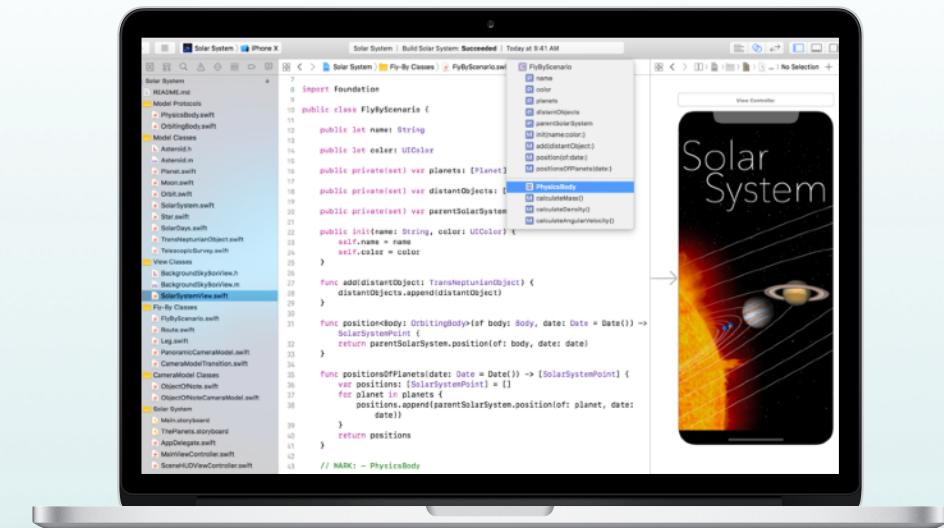
Antes de la primera reunión, asegúrate de que tenéis lo siguiente:

- **Mac.** Los estudiantes necesitarán ordenadores Mac con macOS Mojave o High Sierra. Lo mejor es que cada uno tenga su propio equipo, pero también pueden compartir y programar juntos.
- **Xcode.** Esta app gratuita de Apple para Mac se utiliza para crear todas las apps del Mac, y también las de iOS. Incluye todas las herramientas necesarias para crear apps magníficas. Xcode 9 es compatible con Swift 4, y Xcode 10 es compatible con Swift 4.2.
- **Intro to App Development with Swift (Introducción al desarrollo de apps con Swift).** Este recurso gratuito de Apple guía a los principiantes en ocho proyectos de programación.
- **Keynote.** Usaréis la app Keynote del iPad para hacer los prototipos.
- **Materiales del Swift Coding Club.**

3. Trazá un plan.

Ten en cuenta lo siguiente:

- ¿Quiénes forman parte del club? ¿Qué cosas les interesan? ¿Tienen experiencia en programación o son principiantes?
- ¿Con qué frecuencia se va a reunir el club? Si estás planificando un campamento de verano, ¿cuántas horas vais a dedicar a las actividades de programación?
- ¿Qué tecnología está a disposición del club?
- ¿Cuáles son los objetivos del club?





4. Que corra la voz.

Da a conocer tu Swift Coding Club. Aquí tienes algunas ideas y recursos para atraer a nuevos miembros a tu club:

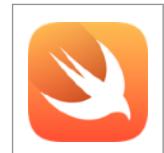
- **Anuncia el club.** Usa el correo electrónico, redes sociales, internet, folletos y el boca a boca para que la gente conozca tu club.
- **Organiza una reunión informativa.** Pregunta a las personas que estén interesadas en formar parte del club qué cosas les interesan y qué tipos de apps les gustaría crear. Hablad sobre organizar una presentación de diseño de apps y cómo pueden participar los miembros. También puedes subir a internet un pequeño vídeo sobre el club.

Los siguientes materiales pueden ayudarte a promocionar y personalizar tu Swift Coding Club:

- **Carteles.** [Descarga esta plantilla gratuita](#) y personalízala para crear tu propio cartel. Imprímelo y exponlo, o haz un cartel digital para subirlo a internet. No olvides indicar cuándo y dónde se reúne el club, y cómo apuntarse.
- **Pegatinas y camisetas.** Usa estas [pegatinas del Swift Coding Club](#) para promocionar tu club. Las camisetas son muy útiles para identificar a quienes participan en los eventos de presentación de las apps. Descarga la [plantilla para camisetas del Swift Coding Club](#) y haz camisetas para los miembros.



Cartel del Swift Coding Club



Pegatina del Swift Coding Club



Camiseta del Swift Coding Club



Aprende y diseña

Los materiales del club están pensados para ayudarte a combinar proyectos de programación y diseño de apps. También puedes añadir sesiones sobre cosas que interesen a los miembros del club. Abajo tienes un ejemplo de planificación para 30 sesiones de una hora.



Puedes añadir sesiones para ampliar los proyectos de programación y diseño de apps, por ejemplo, explorar la realidad aumentada y la realidad virtual, u organizar charlas con diseñadores de apps locales. Para inspirar una lluvia de ideas sobre el diseño de apps, también puedes traer invitados especiales u organizar excursiones.

Consejos para responsables del club



Monta un equipo de dirección. Si tienes un equipo de personas que te ayuden a dirigir el club, todo será mucho más fácil y divertido. ¿Quién tiene dotes de mando? Estudia la posibilidad de designar responsables de coordinar los eventos, la programación, el diseño de apps, etc.

Aprended juntos. Los responsables de un club no tienen por qué saberlo todo. Ayuda a los miembros a investigar por su cuenta y mejorar sus habilidades de resolución de problemas, y animalos a ayudar a los demás.

Presume. Un evento de presentación de apps es una gran oportunidad para promocionar el club, vuestras ideas para apps y vuestro talento para programar entre los amigos, familiares, profesores y la comunidad en general. Incluso puede servir para incorporar a nuevos miembros. En la [página 11](#) encontrarás consejos para organizar tu propia presentación de apps.



Comparte ideas. A algunos miembros les interesarán hacer juegos. Otros igual quieren crear apps para ayudar a la gente, aprender Swift o controlar robots. Busca formas para que los miembros colaboren en proyectos que les interesen.

Júntalos. A veces los miembros más aventajados pueden dejar atrás a los rezagados. Procura emparejar a los primeros con los segundos para que programen juntos. ¡Enseñar a alguien es una forma estupenda de aprender!



Proyectos de desarrollo de apps

Xcode
App Development Projects

Los ocho proyectos explican al principiante los aspectos básicos de los conceptos de programación y del desarrollo de apps. Cada proyecto toma el anterior como punto de partida y termina con la creación de una app funcional, centrándose en diferentes aspectos, de modo que al final los estudiantes tendrán los conocimientos necesarios para crear sus propias apps.

SinglePhoto

Prerequisites: None
Time: 15 hours
See there: Lesson 6: First App

Get how easy it is to build your very first app with SinglePhoto. You'll build an app that not only shows something—the photo—but also reacts when a user taps the screen. This simple app introduction is the basis of many apps, so it will come in handy as you expand your coding and app development skills with Swift. By the end, you'll also have built yourself with Xcode, Interface Builder, and Simulator, and learned how to use them together to build all your own apps.

No hace falta que los miembros realicen los ocho proyectos. El objetivo es presentarles los distintos aspectos de las apps y motivarlos para que diseñen las suyas propias.

AnimalSounds

Prerequisites: ElementaCook app
Time: 1 hour
See there: Lesson 10: Adaptive User Interfaces

As the apps you build become more complex, you may wonder, “Do I have to write all the code myself? Haven’t someone written this code before?” In AnimalSounds, you’ll use some previously created code that does all the hard work of playing music and displaying it on the screen of your app. Then, you’ll go back to Xcode and customize that code to create your most advanced user interfaces yet.

Los miembros necesitarán el curso Intro to App Development with Swift (Introducción al desarrollo de apps con Swift) para hacer los proyectos. ¡Los requisitos son imprescindibles!

Everyone Can Code
Intro to App Development with Swift

[Descargar Intro to App Development with Swift \(Introducción al desarrollo de apps con Swift\) En inglés >](#)

Everyone Can Code
Intro to App Development with Swift Teacher Guide

¿Necesitas más información o quieres profundizar en la cuestión?
[Descargar la guía de profesores de Intro to App Development with Swift \(Introducción al desarrollo de apps con Swift\) En inglés >](#)

Consejos para aprender con Xcode



Consulta la consola para hacer la depuración. La consola puede contener información útil sobre el error. La marca roja muestra la línea del código en la que se ha producido el error.

Detente y piensa. Los errores son inevitables. Detente y piensa en el problema. ¿Cuáles son los síntomas? ¿Funcionaba bien hasta [X]?

No hay una única forma de escribir código. Conviene que los miembros se revisen el código entre ellos, den su opinión y se ayuden con la depuración.

Monta un puesto de ayuda. Dispón un espacio donde los expertos del club puedan ayudar a sus compañeros.



Explora las preferencias de Xcode. Configura la edición de texto y otras preferencias seleccionando Xcode > Preferences en la barra de menús. Puedes añadir cuentas de desarrollador, personalizar la navegación o los tipos de letra, seleccionar determinados comportamientos cuando se producen eventos, etc.

Domina las funciones rápidas de teclado.

Compilar el proyecto: ⌘ B

Compilar y ejecutar el proyecto: ⌘ R

Activar los comentarios en las filas seleccionadas de código: ⌘ /

Desplazar el código seleccionado hacia la izquierda: ⌘ [

Desplazar el código seleccionado hacia la derecha: ⌘]

Ve más allá. Los miembros avanzados pueden hacer el curso App Development with Swift (Introducción al desarrollo de apps con Swift) y centrarse más en los aspectos de programación de sus prototipos de apps en Xcode.



Diario de diseño de apps

The image shows a presentation slide with a light blue background. At the top left is a Twitter logo followed by the text "Swift Coding Club". Below this is a graphic element consisting of a white card tilted at an angle. The card has a small icon of a camera with a play button in the center, followed by the placeholder text "<Your app name>". Below this, in smaller text, is "A little description about your app here". To the right of the card is a black iPhone showing a camera icon on its screen, with the placeholder text "<Your name>" below it. The overall theme is mobile application development.

Los programadores usan este diario de Keynote para aprender sobre las prestaciones de las apps y diseñan una app para resolver un problema de su entorno.

The image shows the front cover of a booklet titled "Keynote for Mac Starter Guide". The title is at the top in a large, bold, black font. Below it, "Starter Guide" is written in a smaller, black font. At the bottom of the cover, there is a teal-colored bar with the text "macOS Sierra" in white. In the top right corner of the cover, there is a small Apple logo. The entire booklet is set against a white background.

Kit de Xcode: Aprende y diseña

Brainstorm

Purpose
Ideas
Audience
Focus
Reiterate

It's important to design with a target audience in mind. Who do you want using your app?



Explore apps from different categories in the App Store. Take a look at the icons, names, and descriptions. Add example images here to keep track of your research.

What have you learned about the audience those apps are meant for?

Did the developers do a good job communicating that?

Judging from the screenshots or preview video, do you think the apps are appropriate for their intended audiences?

En equipos pequeños, realizan una lluvia de ideas y planifican su app, y después crean un prototipo funcional en Keynote.

Prototype

Design
Storyboard
Build

Design

What do the main screens of your app look like? What features appear where? Take a look at your design mood board again and mock up a few screens. You can use Keynote or a drawing app, or draw on index cards to create each screen. Take screenshots or photos of your images and add them to the app you'll build your prototype in.

Tip: Not feeling artsy? Use existing apps as inspiration. Take screenshots of app screens that have great features, add them as a layer or template, and customize them.

 GD Green - Quick start - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit	 gg green - Quick start - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit	 gg green - Quick start - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit	 Virtual Forest - Quick start - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit	 Finance Tracker - Quick start - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit	 Travel Log - Quick start - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit	 Challenge Log - Quick start - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit
Home Screen - Simple - Minimal - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit						
Data Setup - Quick easy data - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit						
Data Organization - Quick visual tools - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit						
Virtual Forest - Quick start - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit						
Finance Tracker - Quick start - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit						
Travel Log - Quick start - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit						
Challenge Log - Quick start - Simple interface - Quick setup - Step-by-step - Detailed - Clean - Minimal - No ads UI Kit						

El diario muestra a los programadores el proceso de evaluación de sus diseños y el desarrollo iterativo de sus prototipos, y emula cómo se diseña una app a nivel profesional.

Los miembros crean un vídeo o una presentación de tres minutos de su app y celebran su trabajo con un evento en el que dan a conocer su diseño.

Celebra



Presentación del diseño de apps

El proceso de diseño de apps y la presentación son excelentes oportunidades para establecer contacto con la comunidad local y explorar el potencial de las apps para resolver problemas contemporáneos. La presentación también es la forma perfecta de exhibir el talento de los miembros del club.

1. Planifica el gran evento. Pon una fecha para la presentación e invita a estudiantes, docentes, padres y madres, y demás miembros de la comunidad.

Asigna tiempo a cada equipo para que presente su app y realiza una breve ronda de ruegos y preguntas. Si tienes un grupo numeroso, puedes dividir el club en dos fases, y así también podrán ver las presentaciones de sus compañeros.

Si quieres, puedes cerrar el evento con un divertido pase de diapositivas de fotos hechas durante las sesiones del club.

2. Crea premios. La competencia sana puede ser una gran fuente de motivación. Inspira a los miembros del club ofreciendo premios que reconozcan talentos concretos relacionados con el diseño de apps. Por ejemplo:

- Mejor Ingeniería
- Mejor Innovación
- Mejor Diseño
- Mejor Presentación Comercial

También podrías animar al público a participar con el Premio del Público.



Puedes descargar y modificar este [certificado](#) para los distintos premios.

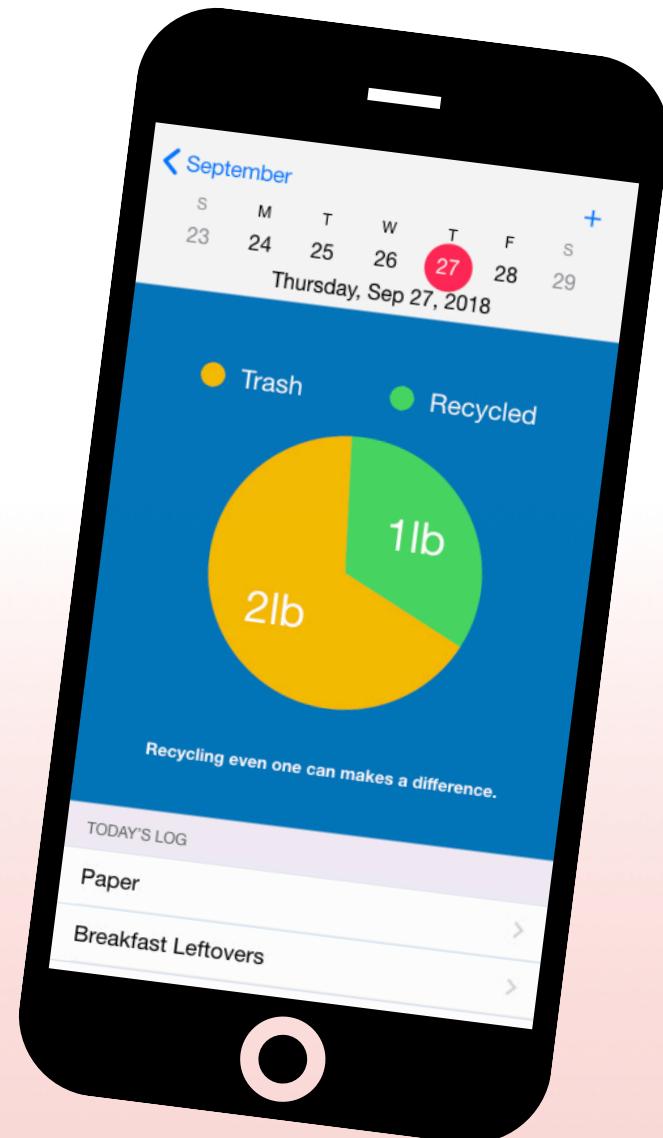


3. Selecciona al jurado y los mentores. Los miembros del jurado y los mentores pueden ser docentes o empleados del centro, estudiantes con experiencia en programación, especialistas del sector del desarrollo o el diseño, integrantes de la junta escolar, personalidades locales o personas a las que pueda beneficiar la nueva app.

No es necesario que el jurado espere hasta la presentación para reunirse con el club. Si te parece, puedes traerlos como invitados especiales para que comparten sus conocimientos cuando los estudiantes estén trabajando en la lluvia de ideas o la planificación del diseño de su app.

4. Elige un ganador. El jurado puede usar la [rúbrica](#) de la siguiente página para evaluar las presentaciones y ofrecer comentarios. También puedes compartir la rúbrica con los programadores antes de la presentación como parte de la fase de evaluación del proceso de diseño de su app.

5. Comparte e inspira. Puedes grabar las presentaciones de las apps. Compártelas con la comunidad local y crea un vídeo con los momentos destacados para inspirar a futuros miembros del club.



Rúbrica de evaluación

[Descargar >](#)

Categoría	Principiante (1 punto)	Intermedio (2 puntos)	Avanzado (3 puntos)	Experto (4 puntos)	Puntos
Contenido de la presentación	Comparte información básica, como el objetivo y los destinatarios.	Da una explicación clara sobre la finalidad de la app, su diseño y cómo aborda las necesidades del usuario.	Da una explicación clara y completa del problema que pretende resolver, la demanda del mercado, el público objetivo y cómo se ha diseñado la app para dar respuesta a las necesidades del usuario.	Hace una presentación convincente con pruebas que demuestran que la app cumple, supera o redefine las expectativas del usuario.	
Exposición de la presentación	Es informativa; una persona del equipo hace la presentación.	Transmite seguridad y entusiasmo; en la presentación intervienen varias personas.	Es interesante y utiliza elementos visuales para ilustrar la historia; el equipo muestra las contribuciones de cada persona.	Ofrece una narración creativa y fácil de recordar; elementos visuales atractivos; transiciones fluidas entre los ponentes del equipo.	
Interfaz de usuario	Pantallas coherentes que contribuyen al objetivo de la app.	Diseño funcional y claro con elementos que resultan familiares al usuario; el prototipo es compatible con las tareas básicas.	Diseño elegante, conciso y agradable con un buen uso del color, la disposición y la legibilidad; el prototipo permite al usuario entender la navegación.	El diseño invita al usuario a interactuar con el contenido; el prototipo utiliza animación, color y disposición de elementos para crear una experiencia fluida y atractiva.	
Experiencia del usuario	Intención clara; los usuarios pueden lograr uno o más objetivos.	Navegación uniforme y estándar; recorrido intuitivo por el contenido de la app.	Adaptable a las necesidades del usuario; tiene en cuenta la accesibilidad, la privacidad y la seguridad.	Innovadora, sorprendente y agradable; ofrece a los usuarios un nuevo tipo de experiencia que se diferencia del resto.	
Conceptos de programación	Cierta conexión entre la funcionalidad de la app y el código.	Explicación sobre cómo los conceptos generales de programación (los tipos de datos, la lógica condicional o los eventos táctiles) se relacionan con la app.	Descripción de las tareas de programación necesarias para desarrollar la app; demostración de cómo el código hace posible la funcionalidad.	Explicación de la arquitectura de la app, la estructura de los datos, los algoritmos y las prestaciones; debate sobre la toma de decisiones para desarrollar este enfoque.	
Comentarios técnicos (opcional) <i>Para prototipos de apps funcionales en Xcode. Los miembros del jurado deben conocer las recomendaciones de desarrollo para iOS y Swift.</i>	El código Swift se ejecuta en ejemplos específicos; el código es básico, sin abstracción.	El código se ejecuta sin errores en todos los casos; el código es básico, con algunos casos de abstracción.	El código está organizado claramente con la nomenclatura de Swift; hay un elevado nivel de abstracción; sigue las directrices de iOS.	El código está bien documentado con comentarios; uso eficaz de las prestaciones de Swift; se sigue una organización, como el patrón modelo-vista-controlador.	
Comentarios:					0 Punt. total



Swift Coding Club

Xcode

Certificado de finalización

Concedido a

Por

Firma

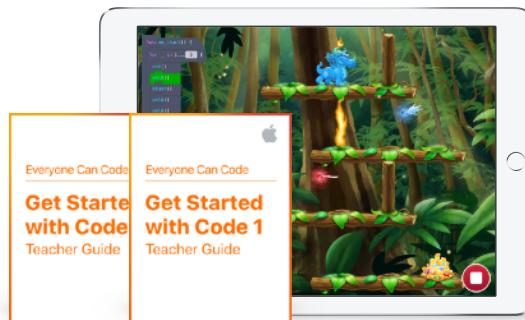
Fecha

Ve más allá

El Swift Coding Club no es más que el principio de tu aventura en el mundo de la programación. El currículo de «Programación para todos» incluye recursos divertidos y de apoyo para pasar de aprender nociones de programación básicas en el iPad a desarrollar apps de verdad en el Mac. Incluso hay certificación en desarrollo de apps con Swift para los estudiantes que han completado el curso correspondiente.

Y puedes ir más allá de las actividades del club. Las completas guías para docentes también permiten llevar la programación a las aulas, con lecciones paso a paso que siguen el currículo desde preescolar hasta la universidad.

[Ver todos los recursos de «Programación para todos» >](#)



[Más información sobre el currículo de Empezar a programar >](#)

[Más información sobre el currículo de Swift Playgrounds >](#)

[Más información sobre el currículo de App Development with Swift \(Desarrollo de apps con Swift\) >](#)



© 2018 Apple Inc. Todos los derechos reservados. Apple, el logotipo de Apple, AirDrop, iPad, Keynote, Mac, macOS y Xcode son marcas comerciales de Apple Inc., registradas en EE. UU. y en otros países. Swift y Swift Playgrounds son marcas comerciales de Apple Inc. Los demás nombres de productos y empresas mencionados en este documento son marcas comerciales de sus respectivas empresas. Noviembre de 2018