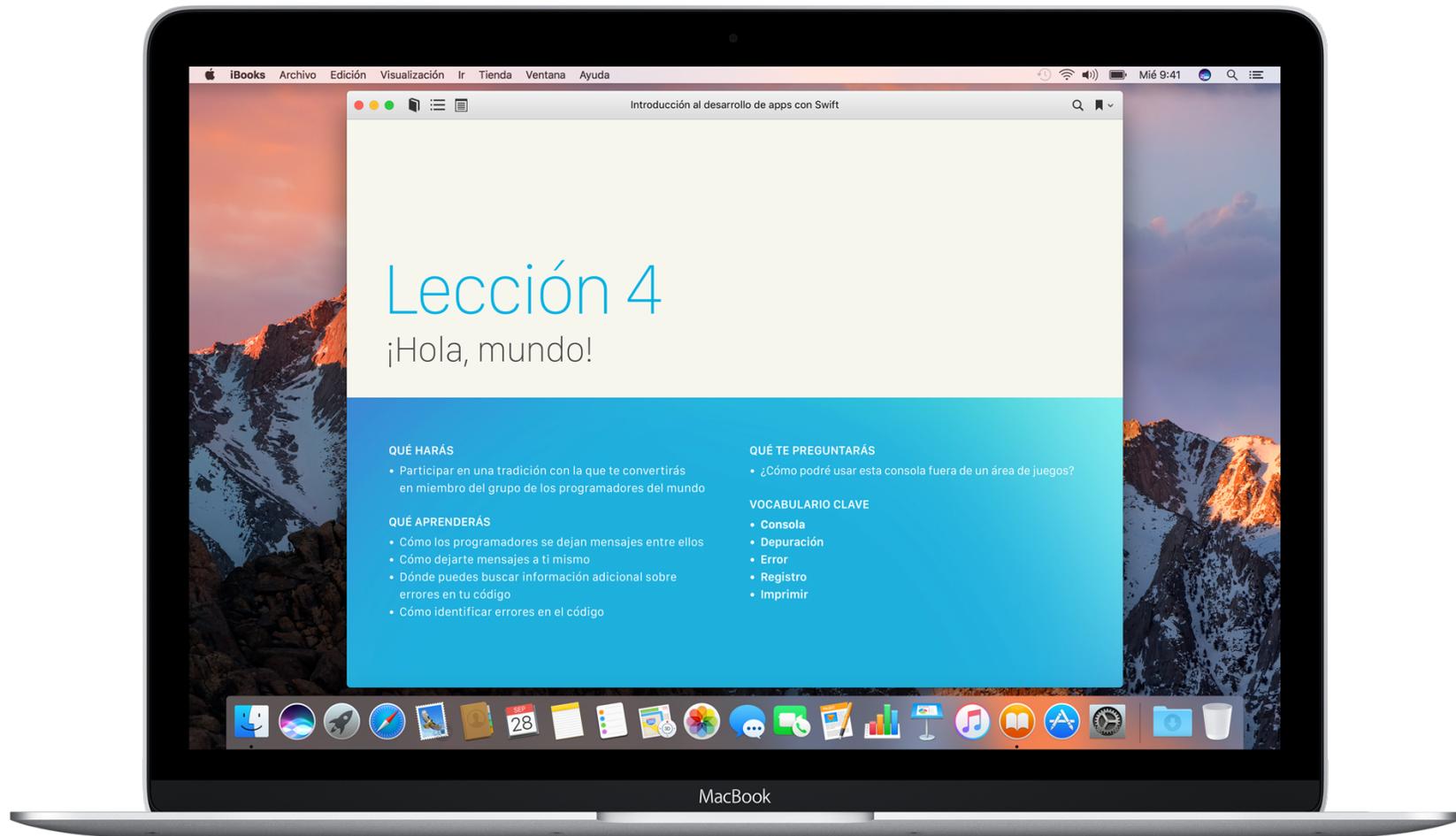




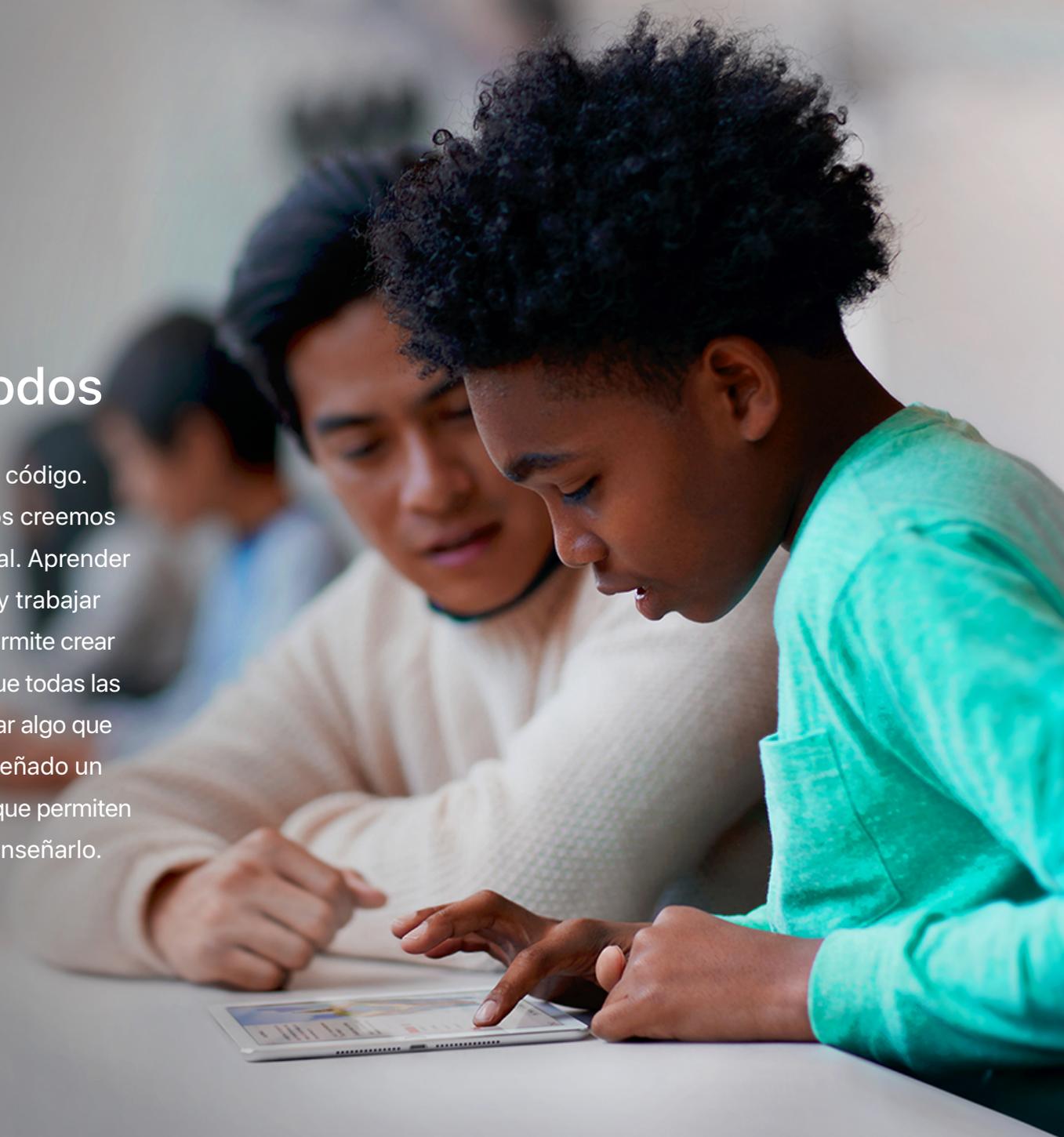
Guía del plan de estudios de Desarrollo de apps con Swift

Septiembre de 2017



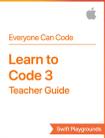
Programación para todos

La tecnología tiene un lenguaje, que se llama código. El código se utiliza para programar, y nosotros creemos que la programación es una habilidad esencial. Aprender a programar te enseña a resolver problemas y trabajar en conjunto de manera creativa. Además, te permite crear apps que hacen realidad tus ideas. Creemos que todas las personas deberían tener la oportunidad de crear algo que pueda cambiar el mundo. Por eso, hemos diseñado un nuevo programa con recursos y herramientas que permiten a cualquier persona aprenderlo, escribirlo y enseñarlo.



Plan de estudios de Programación para todos

Programación para todos incluye una variedad de recursos para que los estudiantes sin experiencia en programación aprendan a crear sus primeras apps. En la tabla que aparece a continuación, se proporciona una descripción general de todos los recursos gratuitos de enseñanza y aprendizaje disponibles.

Plan de estudios	Dispositivo	Audiencia	App	Requisitos previos	Descripción general	Materiales de aprendizaje	Recursos de apoyo	Número de horas de lección incluidas
		Desde la educación inicial hasta el segundo grado		Ninguno	Comienza a pensar como un programador explorando, en la práctica, conceptos de programación mediante apps de base visual.	<ul style="list-style-type: none"> Lecciones de la app codeSpark Academy Curso de Cadete Espacial de Tynker 	<ul style="list-style-type: none"> Empezar a programar 1: Guía para profesores 	30 horas, en las que se incluyen las lecciones de las apps y la Guía para profesores
		Desde el tercer grado hasta el quinto grado		Ninguno	Explora los conceptos fundamentales de programación y practica cómo es pensar como un programador mediante apps de base visual.	<ul style="list-style-type: none"> Curso Hechizos de dragón de Tynker 	<ul style="list-style-type: none"> Empezar a programar 2: Guía para profesores 	36 horas, en las que se incluyen las lecciones de las apps y la Guía para profesores
		Escuela secundaria en adelante		Ninguno	Aprende conceptos fundamentales de programación utilizando código Swift real.	<ul style="list-style-type: none"> App Swift Playgrounds Lecciones de Aprender a programar 1 y 2 Curso de iTunes U 	<ul style="list-style-type: none"> Aprender a programar 1 y 2: Guía para profesores Distintivos de Swift Playgrounds de Apple Teacher Learning Center 	Hasta 85 horas, en las que se incluyen las lecciones de Aprender a programar 1 y 2, y la Guía para profesores
		Escuela secundaria en adelante		Aprender a programar 1 y 2	Amplía tus habilidades de programación y comienza a pensar como un desarrollador profesional de apps.	<ul style="list-style-type: none"> App Swift Playgrounds Lecciones de Aprender a programar 3 	<ul style="list-style-type: none"> Aprender a programar 3: Guía para profesores 	Hasta 45 horas, en las que se incluyen las lecciones de Aprender a programar 3 y la Guía para profesores
		Escuela media y superior en adelante		Ninguno	Adquiere experiencia práctica con las herramientas, las técnicas y los conceptos que se necesitan para crear una app de iOS básica desde cero.	Libro y archivos de proyectos de Introducción al desarrollo de apps con Swift	<ul style="list-style-type: none"> Introducción al desarrollo de apps con Swift: Guía para profesores 	90 horas
		Escuela media y superior en adelante		Ninguno	Construye una base sólida en el uso de Swift, de UIKit y de redes mediante actividades prácticas y proyectos guiados. Los estudiantes podrán crear una app con diseño propio al final de este curso.	Libro y archivos de proyectos de Desarrollo de apps con Swift	<ul style="list-style-type: none"> Desarrollo de apps con Swift: Guía para profesores 	180 horas

Descripción general

Los planes de estudio de Introducción al desarrollo de apps con Swift y Desarrollo de apps con Swift están diseñados para enseñarles a estudiantes que están cursando la escuela media o superior y que tienen poca o ninguna experiencia con la programación, cómo ser desarrolladores de apps, capaces de darle vida a sus creaciones.

En el curso Introducción al desarrollo de apps con Swift, se les presenta a los estudiantes el mundo del desarrollo de apps y los conocimientos básicos de Swift y Xcode. Este curso culmina con un proyecto final en el cual los estudiantes pueden elegir, de entre dos opciones, una app de iOS básica para crear.

En el curso Desarrollo de apps con Swift, los estudiantes cubren temas más avanzados, tanto si son nuevos en el campo de programación como si quieren expandir sus habilidades. Si ya están familiarizados con el desarrollo de apps para iOS con Swift y Xcode, pueden pasar rápidamente por las lecciones o ir directo a las actividades prácticas, donde pueden crear proyectos pequeños y probar el código que crearon en los playgrounds. Al final del curso, podrán crear una app funcional completa, totalmente diseñada por ellos.

Primera app



5.1 Proyecto nuevo

Ahora que te sientes más a gusto con los playgrounds, tal vez te preguntes cómo compilar una app que puedas usar en tu dispositivo iOS o, incluso, en tu Apple Watch. Muchos elementos distintos deben funcionar en conjunto para que una app se ejecute, y Xcode es la mejor herramienta para unirlos.

En esta lección de tres partes, compilarás SinglePhoto, una sencilla app de iOS que muestra una única foto. En el primer ejercicio, crearás un proyecto de app desde cero. Luego, utilizarás Xcode para explorar el proyecto y aprender a navegar por el entorno de programación.

Puedes personalizar todas las partes de la app, desde el ícono en la pantalla de inicio hasta la forma en que se comportan los botones. Xcode incluye paneles y controles que muestran la variedad de opciones que tienes disponibles. Practicarás cómo usar Interface Builder de Xcode para continuar personalizando tu primera app.

En el último paso, agregarás una imagen a tu proyecto y editarás la interfaz de usuario. También verás una introducción a Interface Builder, un potente componente de Xcode donde creas la interfaz de usuario de la app. Al término del segundo ejercicio, tu app tendrá esta apariencia, pero mostrará la foto que tú elijas.

Volver a la página 80 Proyecto nuevo 25

Lesson 1.8 Interface Builder Basics

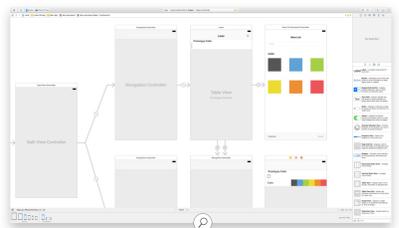
The best way to learn the basics of Interface Builder is to dive into Xcode and explore some of its features. Start by creating a new iOS project using the Single View Application template. Name the project "IBBasics".

STORYBOARDS

Interface Builder opens whenever you select an XIB file (.xib) or a storyboard file (.storyboard) from the project navigator.

An XIB file contains the user interface for a single visual element, such as a full-screen view, a table view cell, or a custom UI control. XIBs were used more heavily before the introduction of storyboards. They're still a useful format in certain situations, but this lesson will focus on storyboards.

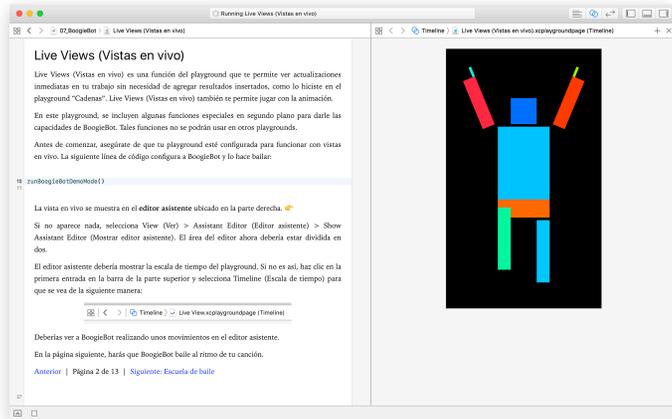
In contrast with an XIB, a storyboard file includes many pieces of the interface, defining the layout of one or many screens as well as the progression from one screen to another. As a developer, you'll find that the ability to see multiple screens at once will help you understand the flow within your app.



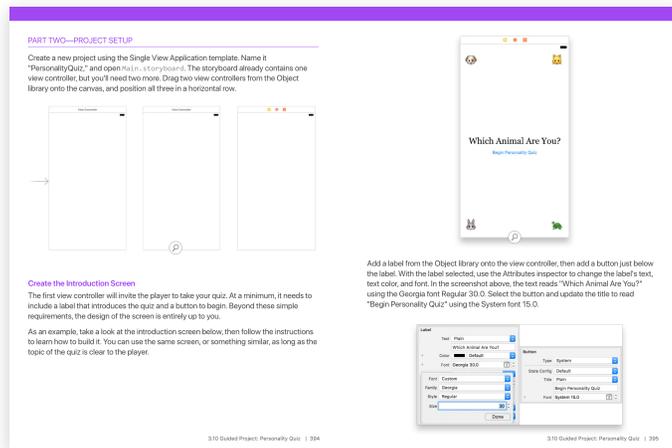
78 1.8 Interface Builder Basics | 79

Características principales

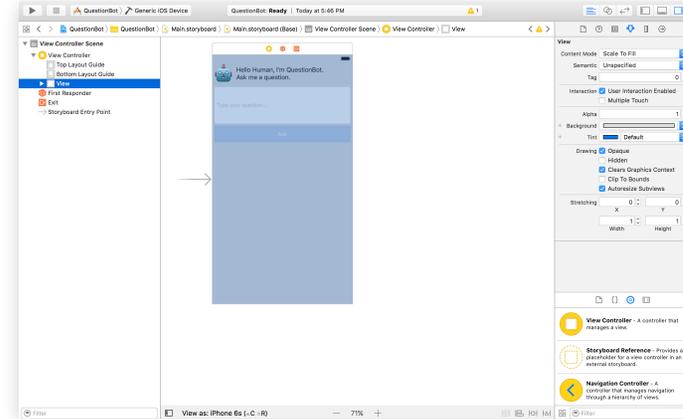
Playgrounds. Los estudiantes aprenden conceptos de programación a medida que escriben código en playgrounds: entornos de programación interactivos que les permiten experimentar con código y ver los resultados al instante.



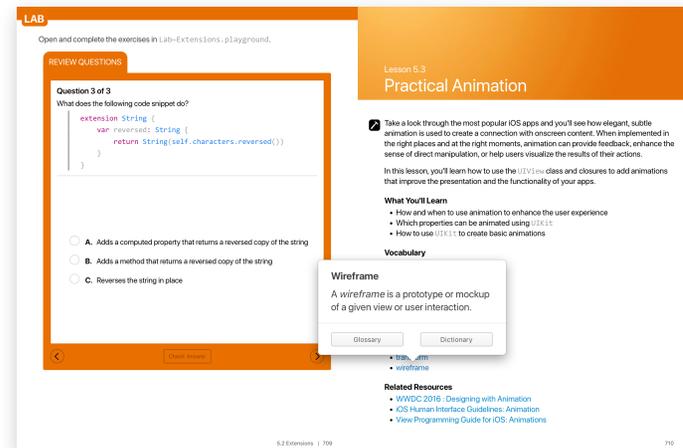
Instrucciones paso a paso. Instrucciones detalladas con imágenes y videos que guían a los estudiantes a través de todos los pasos necesarios para crear una app en Xcode.



Proyectos de ejemplo. Con los archivos de proyecto incluidos, los estudiantes pueden probar ciertas partes del código sin tener que crear una app completa desde el principio.



Herramientas de estudio. Los estudiantes pueden comprobar que han comprendido y aplicar lo que han aprendido con las preguntas de repaso, el vocabulario clave, enlaces a la documentación y mucho más.



Resumen del curso

Introducción al desarrollo de apps con Swift

Este curso de introducción de un semestre está diseñado para ayudar a los estudiantes a crear una base sólida de conocimientos básicos de programación con el lenguaje Swift. Los estudiantes adquirirán experiencia práctica con las herramientas, las técnicas y los conceptos que se necesitan para crear una app de iOS básica.

Las lecciones de Diseño de apps guían a los estudiantes durante el proceso de diseño de una app, en el que se incluyen la propuesta de ideas, la planificación, la preparación de un prototipo y la evaluación de una app diseñada por ellos. Aunque posiblemente todavía no tengan las habilidades para crear la app, el trabajo que hayan realizado con el prototipo los preparará para desarrollarla en el futuro.

Lección 1: Aspectos básicos de los playgrounds. Los estudiantes se familiarizan con el entorno interactivo de los playgrounds.

Lección 2: Nombres e identificadores. Los estudiantes exploran los pasos fundamentales para resolver problemas utilizando nombres e identificadores útiles.

Lección 3: Cadenas. Los estudiantes aprenden el concepto de cadenas y la interpolación de cadenas.

Lección 4: ¡Hola, mundo! Los estudiantes reciben la bienvenida a la tradición de la programación, aprenden a personalizar el entorno de Xcode y a depurar.

Lección 5 : Primera app. Los estudiantes crean su primera app con Xcode y muestran su trabajo en un simulador de iOS.

Lección 6: Funciones. Los estudiantes descubren por qué las funciones son tan poderosas, ya que combinan pasos detallados para crear una definición que se puede usar una y otra vez.

Lección 7: BoogieBot. Los estudiantes aplican su conocimiento de funciones para controlar un robot bailarín animado dentro del playground.

Lección 8: Constantes y variables. Los estudiantes expanden su comprensión de los nombres al aprender formalmente los conceptos de constantes y variables.

Lección 9: Tipos. Los estudiantes se familiarizan con las bases de Swift al examinar el sistema de tipos, desde la biblioteca estándar de Swift hasta tipos personalizados.

Lección 10: Parámetros y resultados. Los estudiantes expanden su conocimiento sobre funciones al aprender acerca de los parámetros y los valores devueltos para que las funciones sean más flexibles y potentes.

Lección 11: Toma de decisiones. Los estudiantes aprenden a tomar decisiones en código con instrucciones condicionales "if/else", los valores Bool true y false, y operadores de comparación.

<p>APLICAR LOS CONCEPTOS (25 minutos)</p> <p>En la lección, usamos la analogía de un artista que crea un cuadro para ayudar a los estudiantes a diferenciar los cuatro tipos de funciones. Comenten los cuatro tipo de funciones:</p> <ol style="list-style-type: none"> 1. ❌ Parámetros, ❌ valor devuelto <code>paintPicture()</code> Funciona por sí solo y no devuelve un valor. 2. ✅ Parámetros, ❌ valor devuelto <code>paintPicture(width: Int, height: Int, dominantColor: UIColor)</code> Realiza tareas que cambian según los argumentos, pero no devuelve un valor. 3. ❌ Parámetros, ✅ valor devuelto <code>paintPicture() -> Painting</code> No requiere información, pero devuelve un valor. 4. ✅ Parámetros, ✅ valor devuelto <code>paintPicture(width: Int, height: Int, dominantColor: UIColor) -> Painting</code> Acepta información y devuelve un valor. 	<p>REVISAR Y COMENTAR (de 10 a 20 minutos)</p> <p>Revisa y comenta las preguntas de reflexión de la guía para estudiantes.</p> <ul style="list-style-type: none"> • Ahora puedes crear funciones que acepten información y la usen en sus tareas. Pero las tareas que se realizan siguen siendo las mismas. ¿Qué sucedería si pudieras realizar una tarea diferente que dependa de la información que se especifica? • ¿Qué procesos y actividades de la vida real se te ocurren que se adaptan a las diferentes maneras en las que puedes definir una función (con o sin parámetros, con o sin un tipo de devolución)? Estos son algunos ejemplos: <ul style="list-style-type: none"> – <code>func turnOffOven()</code> (Apagar el horno no requiere ningún parámetro, y esta acción no devuelve ningún valor). – <code>func preheatOven(temperature: Int)</code> (Precalentar el horno requiere una temperatura como parámetro para empezar el proceso de precalentado). – <code>func bakeCookies() -> [Cookie]</code> (Hornear galletas no requiere ningún parámetro y devuelve un producto dulce). – <code>func bake(ingredients: [Ingredient]) -> [BakedGood]</code> (La repostería general con el horno requiere una lista de ingredientes y devuelve una tanda de productos horneados deliciosos).
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Lección 10 | Parámetros y resultados 52

La Guía para profesores incluye actividades de extensión adicionales, preguntas para debatir y actividades en el diario de la app que los estudiantes deberán escribir durante el semestre.

Resumen del curso (continuación)

Lección 12: Instancias, métodos y propiedades. Los estudiantes profundizan su conocimiento sobre tipos al explorar los métodos y las propiedades que componen una instancia de ese tipo.

Lección 13: QuestionBot. Los estudiantes obtienen experiencia al modificar un proyecto de Xcode existente mediante la creación de nueva lógica para un bot de app que responde diferentes preguntas.

Lección 14: Arreglos y bucles. Los estudiantes descubren cómo crear arreglos y trabajar con ellos agregando y quitando objetos, y aprenden cómo los bucles "for" trabajan con cada uno de los objetos de un arreglo.

Lección 15: Definición de estructuras. Los estudiantes reconocen que suele ser útil agrupar información y funcionalidades relacionadas en un tipo personalizado.

Lección 16: QuestionBot 2. Los estudiantes expanden la app QuestionBot mediante la creación de ChatBot, una app que muestra el historial de la conversación. Examinan el patrón de fuente de los datos y crean un objeto de fuente de datos simple para proporcionar información acerca de objetos "Message" que se visualizan en la vista de lista de mensajes. Los estudiantes practican cómo adjuntar a un arreglo para almacenar mensajes en el objeto de fuente de datos a fin de mantener un historial de la conversación.

Lección 17: Acciones y conectores. Los estudiantes aprenden a crear interfaces de usuario con Interface Builder y a vincular los elementos de la interfaz de usuario con el código mediante acciones y conectores. Practicarán cómo crear conectores para acceder a las propiedades de una vista de interfaz de usuario y cómo crear acciones para responder a la interacción de los usuarios con botones y otros controles.

Lección 18: Interfaces de usuario adaptables. Los estudiantes aprenden un proceso repetible para crear una interfaz de usuario en el dispositivo iPhone de menor tamaño, de modo que se pueda aumentar su escala a todos los tamaños y las orientaciones de los dispositivos iPhone. Explorarán la función de diseño automático, que es el sistema que se usa para establecer restricciones que determinan la ubicación y el tamaño de los elementos de la interfaz de usuario. También utilizarán vistas de pila, que es un objeto especial diseñado para establecer automáticamente las restricciones de diseño automático en función de configuraciones más simples y un sistema tipo cuadrícula. Durante el proceso, crean las apps SimpleCenter, ElementQuiz y AnimalSounds.

Lección 19: Enumeraciones e instrucción "switch". Los estudiantes descubren que las enumeraciones (o "enum") se usan para definir una lista de opciones con nombre. Además aprenden para qué se usan, cómo definir las y las formas más comunes de trabajar con ellas. También aprenderán a usar la instrucción "switch" para ejecutar código específico de manera condicional en función de cualquier opción de una instancia "enum".

Lección 20: Proyecto final. Los estudiantes completan, al menos, una de las dos opciones de proyecto final desde cero. La primera opción es un juego de Piedra, papel o tijeras y la segunda es un generador de memes. Los estudiantes repasan una variedad de conceptos aprendidos durante el curso y crean la interfaz de usuario, los datos representados y los objetos de controlador que componen la app completa.

Lección 21: Diseño de apps. Los estudiantes completan las etapas de un ciclo de diseño que se centra en la preparación de un prototipo, un proceso similar al que realizan los desarrolladores de apps profesionales.

¿Cuál es el próximo paso? Los estudiantes exploran una gran variedad de recursos de desarrollo de apps, desde la página de inicio de Apple Developer hasta la Conferencia Mundial de Desarrolladores (WWDC), donde se presentan los marcos y las herramientas más recientes a fin de crear apps para todas las plataformas de Apple.

Resumen del curso (continuación)

Desarrollo de apps con Swift

Este curso de dos semestres contiene 45 lecciones. Cada una de ellas está diseñada para enseñar una habilidad específica relacionada con Swift o con el desarrollo de apps. Cada tipo de lección tiene un enfoque diferente:

- **Lecciones de Swift.** Estas lecciones se centran en conceptos específicos. Las actividades prácticas para cada lección se presentan en playgrounds para que los estudiantes puedan experimentar con el código y ver los resultados de forma inmediata. Se proporcionan archivos de playgrounds.
- **Lecciones de desarrollo de apps.** Con el objetivo de centrarse en la creación de características específicas para apps de iOS, estas lecciones suelen guiar a los estudiantes paso a paso a través de un pequeño proyecto. Las prácticas ayudan a los estudiantes a aplicar lo que aprendieron a una nueva situación.

Al final de cada una de las primeras cinco unidades, los estudiantes completan proyectos guiados, que incluyen una descripción de las características basadas en el usuario, un plan de proyecto e instrucciones para crear una app completamente funcional. A través de estos proyectos, los estudiantes pueden crear características que les interesan mientras realizan el tipo de trabajo que se espera de un entorno de trabajo de desarrollo de apps. En la última unidad, examinarán cómo diseñar, preparar un prototipo y desarrollar una app diseñada por ellos.

La Guía para profesores incluye consejos para expandir o adaptar las lecciones, aumentar la colaboración y brindarles ayuda a los estudiantes que necesitan asistencia adicional. También incluye presentaciones de Keynote descargables para cada lección, un código de solución para las actividades prácticas y una planilla para evaluar el trabajo de los estudiantes.

Unidad 1: Primeros pasos en el desarrollo de apps. Los estudiantes aprenden conceptos básicos sobre datos, operadores y control de flujo en Swift, así como sobre documentación, depuración, Xcode, creación y ejecución de una app, y sobre Interface Builder. Luego, aplican todo ese conocimiento al proyecto guiado, Linterna, en el que crean una app simple de linterna.

Unidad 2: Introducción a UIKit. Los estudiantes exploran las cadenas, las funciones, las estructuras, las colecciones y los bucles de Swift. También aprenden sobre UIKit (los controles y las vistas del sistema que componen la interfaz de usuario) y cómo mostrar datos con la función de diseño automático y las vistas de pila. Los estudiantes ponen en práctica lo que aprendieron en el proyecto guiado, Tarta de manzana, en el que crean una app para jugar a adivinar palabras.

Unidad 3: Navegación y flujos de trabajo. Los estudiantes descubren cómo crear jerarquías de navegación y flujos de trabajo simples mediante controladores de navegación, controladores de barras de pestañas y transiciones. También examinan dos herramientas potentes de Swift: las opcionales y las enumeraciones. Ponen en práctica este conocimiento con el proyecto guiado, Test de personalidad, una encuesta personalizada que le revela una respuesta divertida al usuario.

Unidad 4: Tablas y persistencia. Los estudiantes aprenden sobre las vistas de desplazamiento, las vistas de tabla y sobre cómo crear pantallas de entrada complejas. También exploran cómo guardar datos y compartirlos con otras apps y cómo trabajar con imágenes en la fototeca del usuario. Usan sus habilidades nuevas en el proyecto guiado, Lista, una app para rastrear tareas que le permite al usuario agregar, editar y eliminar elementos en una interfaz familiar con base de tabla. Los estudiantes pueden personalizar la app para registrar todo tipo de información, como colecciones, tareas o listas de reproducción.

Unidad 5: Trabajo con la Web. Los estudiantes aprenden sobre animaciones, concurrencia y el trabajo con la Web. Aplican lo que aprendieron en el proyecto guiado, Restaurante, una app de menú personalizable que muestra los platos disponibles de un restaurante y que les permite a los usuarios cargar su pedido. Esta app usa un servicio web que les permite a los estudiantes configurar el menú con sus propios elementos y fotos.

Unidad 6: Prototipos y planificación de proyectos. Los estudiantes aprenden a diseñar, preparar un prototipo y desarrollar su propio proyecto. Con el tiempo suficiente, deberán ser capaces de crear un proyecto de forma independiente.

Información adicional

Descargar los recursos de Empezar a programar

- [Tynker](#)
- [codeSpark Academy](#)
- [Empezar a programar 1](#)
- [Empezar a programar 2](#)

Descargar los recursos de Swift Playgrounds

- [Aprender a programar 1 y 2: Curso de iTunes U](#)
- [Aprender a programar 1 y 2: Guía para profesores](#)
- [Aprender a programar 3: Guía para profesores](#)
- [App Swift Playgrounds](#)

Descargar las guías de Desarrollo de apps con Swift

- [Introducción al desarrollo de apps con Swift](#)
- [Introducción al desarrollo de apps con Swift: Guía para profesores](#)
- [Desarrollo de apps con Swift](#)
- [Desarrollo de apps con Swift: Guía para profesores](#)

Recursos adicionales

- Obtén más información sobre el programa [Programación para todos](#).
- Conéctate con otros docentes en los [foros de Apple Developer](#).

Acerca de Swift

Swift es el lenguaje de programación sólido e intuitivo desarrollado por Apple para crear apps, que hace que programar sea más fácil, más flexible y más divertido. Swift no solo es excelente para comenzar a programar, sino que también es absolutamente poderoso. Está diseñado para empezar por escribir el programa más simple, como “¡Hola, mundo!”, hasta diseñar el software más avanzado del mundo.

Obtén más información sobre [Swift](#).

Acerca de Xcode

Xcode es la app de Mac que se utiliza para crear todas las apps para Mac, así como las apps para iOS. Tiene todas las herramientas que necesitas para crear una experiencia sorprendente con las apps. Además, puedes descargarla gratis desde Mac App Store.

Obtén más información sobre [Xcode](#).

Correlación curricular: Introducción al desarrollo de apps con Swift

Aquí se proporciona una correlación preliminar entre Introducción al desarrollo de apps con Swift y los estándares provisionales de 2016 para las ciencias de la computación K–12 de nivel 3A de la Asociación de Profesores de Ciencias de la Computación (CSTA). Una vez que estén finalizados los nuevos estándares, las guías se someterán a una revisión comparativa formal de la CSTA. La correlación abarca los conceptos de algoritmos y programación en los estándares provisionales de 2016 para las ciencias de la computación K–12 de la CSTA.

Estándares para las ciencias de la computación K–12 de la CSTA, nivel 3A (grados 9 y 10)												
Estándar de la CSTA	3A-A-2-1 Diseño de artefactos	3A-A-2-2 Colaboración	3A-A-7-3 Licencia	3A-A-5-4 Respuesta a eventos	3A-A-5-5 Investigación	3A-A-5-6 Conceptos matemáticos	3A-A-4-7 Jerarquía y abstracción	3A-A-4-8 Deconstrucción de problemas	3A-A-4-9 Abstracción	3A-A-3-10 Diseño de algoritmos	3A-A-3-11 Representación y simulación	3A-A-6-12 Depuración
Aspectos básicos de los playgrounds												●
Nombres e identificadores						●		●	●	●		●
Cadenas						●		●	●	●	●	
¡Hola, mundo!								●		●		●
Primera app				●				●				
Funciones						●		●	●	●	●	
Boogie Bot						●		●	●	●	●	
Constantes y variables						●		●	●	●	●	●
Tipos						●		●	●	●	●	●
Parámetros y resultados						●		●	●	●	●	
Toma de decisiones				●		●		●	●	●	●	
Instancias, métodos y propiedades						●		●	●	●	●	
Question Bot				●		●		●	●	●	●	●
Arreglos y bucles						●		●	●	●	●	
Definición de estructuras						●		●	●	●	●	
Question Bot 2				●		●		●	●	●	●	
Acciones y conectores				●		●		●		●		●
Interfaces de usuario adaptables				●		●		●		●		●
Enumeraciones e instrucción "switch"						●		●	●	●	●	
Proyecto final				●		●		●	●	●	●	
Diseño de apps	●	●		●	●							

Correlación curricular: Desarrollo de apps con Swift

Aquí se proporciona una correlación preliminar entre Desarrollo de apps con Swift y los estándares provisionales de 2016 para las ciencias de la computación K–12 de nivel 3A de la Asociación de Profesores de Ciencias de la Computación (CSTA). Una vez que estén finalizados los nuevos estándares, las guías se someterán a una revisión comparativa formal de la CSTA. La correlación abarca los conceptos de algoritmos y programación en los estándares provisionales de 2016 para las ciencias de la computación K–12 de la CSTA.

Correlación de Desarrollo de apps con Swift y los estándares para las ciencias de la computación K–12 de la CSTA, nivel 3A (grados 9 y 10)												
Estándar de la CSTA	3A-A-2-1 Diseño de artefactos	3A-A-2-2 Colaboración	3A-A-7-3 Licencia	3A-A-5-4 Respuesta a eventos	3A-A-5-5 Investigación	3A-A-5-6 Conceptos matemáticos	3A-A-4-7 Jerarquía y abstracción	3A-A-4-8 Deconstrucción de problemas	3A-A-4-9 Abstracción	3A-A-3-10 Diseño de algoritmos	3A-A-3-11 Representación y simulación	3A-A-6-12 Depuración
Unidad 1: Primeros pasos en el desarrollo de apps	●	●	●	●	●	●		●	●	●		●
Unidad 2: Introducción a UIKit	●	●		●		●		●	●	●		●
Unidad 3: Navegación y flujos de trabajo	●	●		●	●	●		●	●	●		●
Unidad 4: Tablas y persistencia	●	●		●		●		●	●	●		●
Unidad 5: Trabajo con la Web	●	●		●	●	●		●	●	●		●
Unidad 6: Prototipos y planificación de proyectos	●	●		●	●	●		●	●	●	●	

Las funcionalidades están sujetas a cambios. Es posible que algunas funcionalidades no estén disponibles en todas las regiones o en todos los idiomas.