

# ÍNDICE

<b>INTRODUCCIÓN.....</b>	<b>XIII</b>
<b>CAPÍTULO 1. EL MUNDO DE LAS APPS .....</b>	<b>1</b>
¿Por qué crear Apps? .....	1
Android vs iOS .....	2
El proceso de crear Apps .....	3
¿Qué necesitamos para crear Apps para iOS? .....	3
¿Qué es iOS? ... ..	5
Tipos de Apps .....	7
Algo más que programar .....	7
<b>CAPÍTULO 2. PREPARÁNDONOS PARA LA BATALLA.....</b>	<b>11</b>
¿Comenzamos a crear Apps? .....	11
Descargar simuladores y documentación.....	12
Crear una cuenta de desarrollador en Apple .....	13
Tu primera App .....	14
Todo lo que debes saber por ahora .....	21
Crear la vista .....	22
Compilar el proyecto.....	26
Crear conexiones.....	29
Crear la lógica.....	34
<b>CAPÍTULO 3. XCODE .....</b>	<b>41</b>
Xcode .....	41
LLVM y Clang .....	41
Secciones del IDE .....	42
Storyboard vs .xib.....	51
Playground .....	52

Autolayout .....	55
Añadiendo restricciones.....	59
Orientaciones de pantalla .....	62
Trait Variations .....	63
Entendiendo Trait Variations .....	64
Aplicando Trait Variations .....	66
Plantillas disponibles.....	72
Configurar nuestro proyecto.....	75
El archivo info.plist.....	77
El target del proyecto.....	80
Algunas funciones útiles.....	86
<b>CAPÍTULO 4. INTRODUCCIÓN A LA POO .....</b>	<b>89</b>
¿Qué es la programación orientada a objetos? .....	89
Objetos y clases.....	90
Abstracción .....	91
Encapsulación .....	91
Herencia .....	92
Polimorfismo .....	92
Pero... todo esto ¿para qué? .....	92
<b>CAPÍTULO 5. SWIFT .....</b>	<b>93</b>
Fundamentos de Swift .....	93
Características .....	94
Variables y constantes .....	95
Operadores básicos.....	97
Practicando con Swift.....	100
Comentarios .....	102
Declarar variables y constantes.....	103
Optionals .....	107
Type Casting o Conversión de tipos .....	111
Tuplas .....	115
Clases .....	116
Estructuras .....	128
Enumeraciones .....	129
Propiedades .....	132
La función print .....	134
Interpolación de strings .....	135
Funciones .....	135

Closures .....	142
Métodos .....	144
Colecciones .....	144
Estructuras de control o control de flujo .....	155
Where .....	170
Defer .....	171
Control de errores.....	172
Generics .....	175
Extensiones .....	176
POP-Programación Orientada a Protocolos .....	178
<b>CAPÍTULO 6. COCOA TOUCH Y FOUNDATION .....</b>	<b>179</b>
Introducción .....	179
Los Frameworks más importantes .....	180
Cocoa Touch .....	182
UIKit .....	182
Core Services .....	189
Foundation .....	190
<b>CAPÍTULO 7. MI MÉTODO PARA CREAR APPS .....</b>	<b>191</b>
Cuatro conceptos clave .....	191
Crear una clase y asignársela a un ViewController.....	194
Patrones de diseño .....	202
El rey de los patrones: MVC .....	202
Crear vistas, crear código y conectar .....	240
Crear objetos por código.....	241
Detrás de la magia .....	242
ARC – Automatic Reference Counting.....	258
<b>CAPÍTULO 8. INTEROPERABILIDAD.....</b>	<b>261</b>
Mezclar Swift con Objective-C .....	261
Integrar Swift en un proyecto de Objective-C.....	261
<b>CAPÍTULO 9. UIKIT.....</b>	<b>263</b>
Introducción .....	263
Los ingredientes para crear fantásticas vistas.....	265
UIButton .....	266
UILabel .....	266
UITextField .....	266

UITextView .....	267
UIAlertController .....	267
UIPickerView .....	273
UIDatePicker .....	284
UISwitch .....	290
UISlider .....	293
UIActivityIndicatorView .....	296
UISegmentedControl .....	296
UIStepper .....	300
UIProgressView .....	305
Hasta aquí UIKit.....	309
<b>CAPÍTULO 10. FOUNDATION .....</b>	<b>311</b>
Foundation - La génesis de Cocoa .....	311
NSObject .....	312
NSString .....	312
NSArray .....	312
NSDictionary .....	313
NSNumber .....	313
Data .....	314
URL .....	314
<b>CAPÍTULO 11. APLICACIONES MULTIVISTA .....</b>	<b>315</b>
Introducción .....	315
UITabBarController .....	315
UITabBarController programando con Swift .....	316
UITabBarController con Storyboard .....	333
UINavigationController .....	342
UINavigationController con Storyboard .....	342
UINavigationController programando con Swift .....	348
<b>CAPÍTULO 12. VISTAS DE TABLA .....</b>	<b>361</b>
Introducción .....	361
UITableView .....	362
UITableViewCell .....	363
Crear una App de vista de tabla .....	364
Crear nuestra celda personalizada .....	383
Poniendo una vista detalle .....	391

<b>CAPÍTULO 13. WEB, EMAIL Y REDES SOCIALES .....</b>	<b>401</b>
Conectar nuestra App .....	401
Diferentes formas de ver una Web .....	402
Mostrar una Web embebida en nuestra App - UIWebView .....	402
Ver una Web embebida en nuestra App - WKWebView .....	416
UIWebView VS WKWebView .....	420
SFSafariViewController .....	421
Enviar un email .....	423
<b>CAPÍTULO 14. PERSISTENCIA DE DATOS LOCAL .....</b>	<b>433</b>
Guardar datos de forma persistente.....	433
UserDefaults .....	433
NotificationCenter.....	454
NSFileManager .....	463
Otros tipos de persistencia .....	471
<b>CAPÍTULO 15. LOCALIZAR NUESTRA APP .....</b>	<b>473</b>
Cómo localizar y adaptar nuestra App .....	473
<b>CAPÍTULO 16. MAPKIT.....</b>	<b>495</b>
Mapas en tu App.....	495
Distintos tipos de mapas .....	495
Ubicar al usuario.....	495
Crear una App con mapa y localización .....	496
Obtener la localización del usuario .....	501
<b>CAPÍTULO 17. CÁMARA .....</b>	<b>507</b>
Usar la cámara .....	507
Guardar fotos .....	508
<b>CAPÍTULO 18. CREAR UNA APLICACIÓN .....</b>	<b>515</b>
Poner las cosas sobre la mesa.....	515
La mejor forma de comerse un elefante .....	515
Ten clara tu idea y focaliza .....	516
Definir la navegación .....	516
Crear las vistas teniendo en cuenta el diseño .....	516

Programar la lógica.....	517
Probar que todo funcione .....	517
<b>CAPÍTULO 19. ¿Y AHORA QUÉ?.....</b>	<b>519</b>
Ya tienes tu aplicación, ¿y? .....	519
La imagen y el título de tu App.....	520
Crear la imagen de tu App.....	521
iTunes Connect.....	523
Subir tu aplicación al App Store.....	538
Seleccionar el Certificado con el que vas a firmar la compilación .....	542
Crear el archivo .IPA.....	544
Recomendaciones finales .....	549
<b>ÍNDICE ANALÍTICO .....</b>	<b>551</b>

# INTRODUCCIÓN

## CÓMO EMPEZÓ TODO

---

En el año 2008 mi primo Raúl me recomendó que probara un nuevo teléfono móvil inteligente que la compañía Apple acababa de lanzar al mercado, él lo había adquirido unos meses antes, y estaba muy contento con su compra.

Yo estaba satisfecho con mi Nokia N70, pero eran tantas las alabanzas que me llegaban de este nuevo dispositivo, que un amante de la tecnología como yo no podía dejar pasar la oportunidad de probar este nuevo smartphone.

Steve Jobs lo había presentado un año antes en una de sus míticas Keynotes: “...de vez en cuando aparece un producto revolucionario que lo cambia todo...”.

Definió a aquel dispositivo como un producto revolucionario, y en aquel momento lo fue, vaya que si lo fue. En aquella presentación Jobs explicó que el dispositivo que estaba a punto de presentar, integraba tres productos en uno: un iPod con una gran pantalla táctil, un teléfono móvil revolucionario, y por último un dispositivo de comunicación por internet sin barreras.

Todo ello en un solo dispositivo que llamaron iPhone.

El iPhone cambió la historia de Apple, y también cambió mi vida.

Compré mi primer iPhone en agosto de 2008, el modelo 3G, que fue el primero en llegar a España, ya que el anterior, no llegó a salir en mi país.

El dispositivo fue una auténtica revolución, si tuviste uno en tus manos sabes de lo que te hablo; ahora es hablar por hablar qué smartphone es mejor o peor, es más una cuestión de gustos o afinidad a una determinada marca, pero por aquel entonces el iPhone supuso un cambio tecnológico importantísimo, nunca antes habíamos pellizcado una foto en nuestro teléfono para hacerla más grande, nunca antes habíamos navegado por internet con tanta libertad como empezamos a hacerlo a raíz del primer iPhone.

Aquel año, Apple también presentó el App Store, una aplicación que serviría para que desarrolladores, grandes y pequeños, pudieran distribuir sus Apps en el iPhone.

El App Store vendría instalado en cada iPhone, de forma que cualquier desarrollador podía enviar sus aplicaciones gratuitas o de pago para poder ser distribuidas.

Muy sencillo, tu creabas una aplicación, la subías al App Store, y si vendías algo, tú como desarrollador ganabas el 70% de la venta, y Apple se quedaba con el 30%, esto a día de hoy no ha cambiado.

Hoy en día la venta de aplicaciones se ha convertido en una industria que genera más ingresos que Hollywood.

Empecé a consumir Apps, la mayoría gratuitas, pero de vez en cuando compraba algunas, y un buen día pensé: “estoy gastando mi dinero en comprar Apps, creo que esto tiene futuro, y me hice la siguiente pregunta: ¿cómo se crearán estas aplicaciones?”.

Y esa pregunta que me hice me ha traído hasta hoy, con experiencia de haber llegado a tener más de 50 aplicaciones en el App Store, de haber sido número 1 en muchos países, de haber obtenido más de 200.000 euros en ingresos por la venta de mis Apps, y lo más importante: haber encontrado mi pasión y el verdadero motor de mi vida profesional.

Nunca había programado antes de crear Apps, pero tuve la firme voluntad de aprender cómo se hacían esas pequeñas aplicaciones con las que tanto disfrutaba en mi iPhone, y como todo es cuestión de actitud, logré mi propósito.

Hoy me sigo haciendo la misma pregunta cada vez que afronto un nuevo proyecto para crear una App: ¿cómo se creará esta aplicación?



Si alguna vez te has hecho esta pregunta, y estoy seguro de que así ha sido, pues de otro modo no estarías leyendo estas líneas, te invito a que sigas leyendo, a que devores este libro con pasión, a que apliques mi método, pues si tú quieres, yo te voy a enseñar a crear Apps para iPhone y iPad.

## ¿Para quién es este libro?

---

Este libro está pensado para cualquier persona que quiera aprender a desarrollar aplicaciones para iPhone y iPad con Swift, con las siguientes condiciones:

- No es necesario tener conocimientos previos de programación
- No se necesita conocer ningún lenguaje de programación

Así que, este libro parte de la base de que no sabes nada.

Por otra parte, y debido a su estructura, si ya tienes conocimientos de programación, conoces algún otro lenguaje de programación que no sea Swift, o incluso si ya has desarrollado aplicaciones para iOS con Objective-C, este libro también es válido para ti. Lo único que debes hacer es repasar el capítulo de Xcode, y empezar a leer el libro por el capítulo 5 (aunque quizá te vendría bien un repaso de todos los temas). Otra opción es ir directamente a lo que te interese, pues como digo, no es necesario seguir una estructura lineal si ya conoces algunos de los temas tratados en el mismo.

Por lo tanto, este libro es para principiantes sin ninguna experiencia previa en creación de Apps o programación, y también para los que ya han hecho sus cosas en este campo, pero quieren aprender Swift, a usar Swift con Cocoa, o simplemente disponer de un manual de consulta de casos prácticos.

Aunque no es estrictamente así, este libro comprende dos secciones bien diferenciadas, por una parte hay teoría, que serían los capítulos 1 al 8, y donde se explican fundamentos y metodología; por otra parte están el resto de capítulos, donde se ve cómo aplicar en la práctica todo lo aprendido en capítulos anteriores. Si bien es cierto que se explican algunos conceptos, es una sección del libro que está más pensada para practicar y aprender a usar las diferentes herramientas y tecnologías que nos van a permitir crear asombrosas aplicaciones. Y digo que no es estrictamente así, porque en todos los capítulos se explican algunos fundamentos de la práctica a realizar.

Probablemente se trate del material escrito y publicado de habla hispana más completo del mercado. Y lo mejor de todo es que no es un “refrito” traducido de otros idiomas. Todo lo aquí expuesto es completamente original, y es fruto de mi experiencia y metodología, que viene siendo probada en cientos de alumnos tanto en cursos presenciales como online en varios años.

Una vez finalizada la lectura de todos los temas, tendrás una sólida base que te permitirá desarrollar aplicaciones con Swift para dispositivos iOS de cualquier complejidad.

## Convenciones

---

Este libro enseña a crear aplicaciones con **Xcode 9**, la última versión estable y publicada por Apple en el momento de escribir estas líneas.

No te preocupes si cuando leas este libro hay una versión más avanzada del IDE, generalmente las modificaciones al nivel que nos encontramos (Iniciación – Intermedio) no nos afectan demasiado.

Así mismo se ha usado **Swift 4** como lenguaje de programación, que es la última versión estable del lenguaje que ofrece Apple a la comunidad de desarrolladores.

Como digo, en la mayoría de casos no debería suponer ningún problema si a la hora de leer este libro, tu versión de Xcode o Swift es posterior; el ritmo de actualizaciones que Apple impone (una vez al año como mínimo) hace muy complicado actualizar el material, aunque en la medida de lo posible se harán actualizaciones y correcciones si fuese necesario.

## Metodología, Swift y usar Swift con Cocoa

---

A continuación veremos un breve resumen del contenido que aprenderás en cada capítulo:

1. **Capítulo 1. El mundo de las Apps:** en este capítulo haremos una breve introducción de conceptos muy generales de lo que es el desarrollo iOS. Así mismo veremos qué necesitamos para empezar a crear Apps para iPhone y iPad.
2. **Capítulo 2. Preparándonos para la batalla:** descargar e instalar Xcode, y cómo crear una cuenta de desarrollador de Apple, es lo que veremos en este capítulo.

3. **Capítulo 3. Xcode:** el software que usaremos para crear Apps. Aprenderemos cómo usar el IDE que Apple nos proporciona para crear nuestras aplicaciones.
4. **Capítulo 4. Introducción a la POO:** no te preocupes si nunca has programado, en este capítulo veremos algunos fundamentos de Programación Orientada a Objetos. Lo imprescindible para empezar a desarrollar en iOS.
5. **Capítulo 5. Swift:** uno de los capítulos más importantes del libro. Veremos el nuevo lenguaje de programación de Apple Swift en su versión 4, la última publicada por Apple en el momento de escribir estas líneas.
6. **Capítulo 6. Cocoa Touch y Foundation:** ¿sabes lo que es un Framework? En el capítulo 7 te lo cuento con detalle. En este capítulo veremos dos de los más importantes y que usaremos muy a menudo.
7. **Capítulo 7. Mi método para crear Apps:** no se trata de aprender un lenguaje, ni de aprender a programar, ni siquiera se trata de saber usar Xcode. Hay que aprender una metodología, adquirir hábitos, conocer trucos, que hagan tu aprendizaje algo llevadero. Con mi método no caerás en la desesperanza de no saber de qué te están hablando.
8. **Capítulo 8. Interoperabilidad:** tanto si ya has creado aplicaciones para iOS con Objective-C, como si no, este capítulo te interesa. Veremos cómo usar Swift con Objective-C y viceversa. Te sorprenderá lo bien que se llevan ambos lenguajes.
9. **Capítulo 9. UIKit:** aprenderás a manejar los objetos para crear fabulosas vistas. Este capítulo es la caja de herramientas que abrirás y usarás cada vez que quieras crear una aplicación.
10. **Capítulo 10. Foundation:** si en el anterior capítulo veíamos las herramientas para crear fantásticas vistas, en este veremos los objetos que nos permitirán hacer que esas vistas piensen y actúen. Aprenderás a usar objetos de Foundation, uno de los Frameworks más antiguos y más importantes del universo Cocoa.
11. **Capítulo 11. Aplicaciones Multivista:** toda aplicación que se precie debe tener más de una vista. Verás algunas opciones disponibles para tal fin. TabBar Controller, Navigation, UINavigationController son, entre otros, controladores de vistas que debes conocer.
12. **Capítulo 12. Vistas de Tabla:** en algún momento deberás usar este objeto. Las vistas de tabla son la mejor forma de presentar información estructurada. Son muy potentes, y en este capítulo aprenderemos a usarlas.

13. **Capítulo 13. Web, email y redes sociales:** este tipo de aspectos no pueden faltar en tu App. Veremos cómo integrar el envío de emails, además aprenderás a mostrar una Web en tu App.
14. **Capítulo 14. Persistencia de datos local:** la persistencia de datos consiste en almacenar datos de tu aplicación en el dispositivo, de forma que cuando el usuario cierre la aplicación por completo, al volver a usarla, pueda recuperar los datos guardados. Aprenderás a usar diferentes métodos de guardado de datos de forma persistente.
15. **Capítulo 15. Localizar nuestra App:** no queremos que tu producto, tu App, sea un producto local. Como mínimo tu aplicación debería estar en dos idiomas, el tuyo nativo y el inglés. En este capítulo aprenderás a traducir tu aplicación a cualquier idioma que quieras.
16. **Capítulo 16. MapKit:** descubrirás cómo implementar mapas en tu aplicación. Para ello aprenderás a usar dos Frameworks que Apple nos proporciona para tal fin: MapKit y CoreLocation.
17. **Capítulo 17. Cámara:** aprende a usar la cámara de los dispositivos iOS. Para hacer fotos y guardar las mismas, o bien tomar una foto del carrito.
18. **Capítulo 18. Crear una aplicación:** en este capítulo te daré algunas pautas que debes tener en cuenta para crear una aplicación.
19. **Capítulo 19. ¿Y ahora qué?:** ya prácticamente has aprendido todo. Incluso puede que hayas terminado alguna aplicación. En este capítulo veremos cuáles son los siguientes pasos a dar. Lo posterior será cómo configurar todo para subir tu aplicación al App Store.

## Cómo leer este libro

---

Mi recomendación es que leas el libro, y en una primera lectura no hagas ningún ejercicio, lee e intenta entender cada concepto, cada explicación de cada línea de código.

En una segunda lectura, haz todos y cada uno de los ejercicios mirando los ejemplos que contienen, asimila de nuevo cada explicación, ya que en esta segunda lectura pondrás los ejercicios en práctica, reforzarás conceptos.

Por último, estaría bien que hicieses cada ejercicio tantas veces como sean necesarias hasta que pudieses realizar cada ejemplo sin mirar mi código.

Sé que no me vas a hacer ni caso, y desde el primer momento vas a copiar el código en Xcode para realizar los ejemplos, no pasa nada, tan solo es mi

recomendación, pero puedes hacer todos los ejercicios desde la primera lectura, y luego repasar conceptos teóricos una vez hayas realizado la práctica.

Bien, toma aire, y esboza una gran sonrisa, vas a convertirte en iOS Developer, vas a adentrarte en el fantástico mundo de las aplicaciones para móviles.

¿Estás preparado? Pues vamos a por ello, que empezamos a crear Apps en nada.

## **Agradecimientos**

---

Muchas gracias a Silvia, mi esposa y compañera de viaje, por hacerse cargo de tantas cosas mientras yo “aporreaba” las teclas de mi Macbook día y noche, gracias de verdad.

Gracias a mi madre, Paloma, y a mi hermana María, por entenderme y animarme a seguir creando.

Gracias a Steve Jobs, gracias, muchísimas gracias por tanta inspiración, pues no es lo que se cuenta o se sabe de él, es lo que imaginamos, lo que queremos crear, gracias en definitiva por tanta inspiración.

Muy agradecido a Apple y a los ingenieros y las personas que hacen realidad el hardware sobre el que construir nuestro software.

Mil gracias a todos los alumnos y alumnas que se formaron de forma presencial en CFE APPS, cómo me gustaría enseñarles de nuevo a crear Apps, dándoles más conocimiento.

Gracias a los integrantes de Campus iOS Online, pues son el motor que hacen que siga avanzando en mis objetivos. Gracias también a los oyentes de mi Podcast y programa “Haciendo Apps”.

## **Sobre el autor**

---

Sergio Becerril es desarrollador iOS desde el año 2010, sus aplicaciones han sido número 1 en más de 80 países, fue pionero en la creación de Apps de entretenimiento, y Apps para ver la televisión en iPhone.

Desde el año 2012 es Instructor iOS, e imparte formación online en las principales plataformas educativas, ocasionalmente ofrece formación presencial en Centros educativos y empresas.

Tiene su propio blog en [www.cfeapps.com](http://www.cfeapps.com) donde enseña a crear Apps a todo el que quiera aprender, puedes encontrar todos sus cursos en su propia plataforma educativa de iOS en [www.campusiosonline.com](http://www.campusiosonline.com).

Además, es el creador del programa y Podcast “Haciendo Apps”, un espacio en formato programa en Youtube y Podcast respectivamente, donde se habla y se enseña todo lo que rodea el desarrollo de aplicaciones móviles.

Vive con su mujer y sus dos hijas en Colmenar Viejo, Madrid.

# 1

# EL MUNDO DE LAS APPS

## ¿POR QUÉ CREAR APPS?

---

Lo bueno de todo esto es que cualquiera desde el salón de su casa puede crear una aplicación y subirla al App Store.

Hay que probar suerte, ver si con tu creación o creaciones, puedes ganar algo de dinero; sin duda querer hacerte millonario puede ser tu principal motivación, sobre todo al principio. Pero luego quizá descubras que crear Apps es algo tan fascinante, y tiene tantas posibilidades, que a lo mejor encuentres algo más importante que la compensación económica, puede que encuentres tu pasión.

He tenido alumnos y alumnas de todo tipo, personas que querían aprender a crear Apps para buscar un trabajo como desarrollador en alguna empresa, otros, sin embargo, tenían una idea de una aplicación y querían desarrollarla y subirla al App Store, también había alumnos que aprendían para establecerse por su cuenta como Freelance y crear aplicaciones para otras personas o empresas, y también he tenido otros que simplemente querían aprender por pura diversión, sin más motivación que ampliar su conocimiento.

Sea cual sea la razón por la que has decidido empezar a crear Apps, es fantástico, y con tu interés y mi ayuda conseguirás tu objetivo, estoy seguro de ello.

En este primer capítulo veremos todo lo que necesitamos para empezar nuestro aprendizaje, así como algunos apuntes generalistas del proceso de crear Apps.

Se trata de un capítulo de fácil digestión, y aunque podría parecer el típico capítulo que uno pueda saltarse, te recomiendo que lo leas, pues tiene cosas muy interesantes.

Cuando uno decide crear Apps, debe tomar una primera decisión:

¿Desarrollo aplicaciones para Android o para iOS?

En este libro vamos a aprender a crear Apps para iPhone y iPad, es decir, para dispositivos iOS, y te aseguro que es la mejor opción para empezar en este mundo, no solo porque crear aplicaciones para iOS es más sencillo, sino que el App Store es un mercado mucho más rentable que el mercado de Google Play.

## **Android vs iOS**

---

Los dispositivos Android publican sus aplicaciones en Google Play, y los dispositivos iOS lo hacen en el App Store.

Cada año se publican estadísticas de los principales markets y hay dos tendencias claras, que cada año se acentúan más: Google Play cada vez tiene más descargas que el App Store, la otra tendencia y es la verdaderamente importante: el App Store supera cada vez más en ingresos a Google Play.

¿Qué quiere decir esto? Pues que si buscas descargas, tu sitio es Google Play, pero si quieres obtener ingresos por la venta de tus aplicaciones, entonces estás en el lugar correcto, pues en el App Store los desarrolladores obtienen más del doble de beneficios que los desarrolladores que publican sus Apps en la tienda de Google.

Y esto es todo lo que escribiré en este libro sobre Android y Google Play.

Hay una web que trimestralmente publica analíticas de los principales mercados de Apps, esta Web es [www.appannie.com](http://www.appannie.com) totalmente recomendable si vas a dedicarte al desarrollo de Apps, pues también ofrece un servicio en el cual te informan todos los días, mediante un email, de las descargas que han tenido tus aplicaciones, y esto de forma completamente gratuita.

App Annie es todo un referente en el desarrollo móvil, y aunque tiene servicios premium, puedes crear una cuenta gratuita y disfrutar de alguno de sus servicios.



## El proceso de crear Apps

---

Para crear aplicaciones lo más importante es tener un objetivo claro.

Si tu idea es crear aplicaciones y distribuirlas en el App Store por tu cuenta, lo más importante es tener una idea, y da igual si esa idea es buena o mala, tú no eres quien para valorar tu idea.

Fíjate qué idea tan absurda hizo millonarios a unos jóvenes, pensaron en un juego que consistía en lanzar pajaritos contra unos cerdos con un tirachinas, pero es que además, los cerdos ni se movían, sí, lógicamente estoy hablando de Angry Birds.

O ese joven que pensó llevar el juego del Scrabble a los smartphones, lo llamó Apalabrados. Diset le compró la licencia para hacer un juego de mesa, y según dicen, la versión de mesa de Apalabrados ¡supera a Scrabble en ventas!

Tu idea será buena hasta que lances tu aplicación y veas si la gente descarga tu App o no.

Lo dijo Picasso, y Steve Jobs era de la misma opinión:

“Los grandes artistas copian, los genios roban”.

Por lo que si no se te ocurre ninguna idea, date una vuelta por el App Store y mira cuál es la tendencia del mercado, y haz aplicaciones iguales o parecidas.

## ¿Qué necesitamos para crear Apps para iOS?

---

Esta pregunta tiene fácil respuesta: ganas.

Bromas aparte, para crear Apps para iOS necesitamos hardware y software específico, además de dominar algunos aspectos cómo:

- **Objective-C o Swift:** es esencial aprender uno de estos lenguajes de programación. Swift es el lenguaje que aprenderemos a usar en este libro, y lo veremos de forma extensa en el capítulo 5.
- **Xcode:** debemos saber manejar con soltura esta herramienta para crear Apps.
- **Cocoa y Cocoa Touch:** dos de los más importantes Frameworks que necesitamos conocer. Un Framework es un conjunto de clases

predefinidas, es decir, un conjunto de objetos que ya han sido creados y podemos reutilizarlos en nuestras aplicaciones. Cocoa y Cocoa Touch son Frameworks en sí mismos. El primero para el entorno de Mac, y el segundo para los dispositivos iOS. No te preocupes por esto ahora, pues lo veremos en detalle un poco más adelante.

- **Patrones de diseño:** es necesario conocer algunos patrones de diseño, que no son más que un conjunto de buenas prácticas para resolver muchos de los problemas con los que nos encontramos cuando estamos programando.

## Hardware

Quizá el mayor impedimento para crear aplicaciones para iPhone y iPad sea que para desarrollar aplicaciones y subirlas al App Store necesitamos un ordenador Mac.

No tiene que ser el Mac de última generación con todos los extras, basta con uno que permita instalar y hacer funcionar la última versión de Xcode.

Para que te hagas una idea, yo compré mi primer Mac, un Macbook Pro de 13", a plazos, lo pagué en 12 meses en el año 2009. Con esto quiero decirte que hay muchas formas de conseguir un ordenador de Apple.

Puedes mirar en eBay, comprobar las estrellas del vendedor, es decir, que sea un vendedor de fiar, y comprar por esta vía. Te aseguro que hay auténticas gangas.

Si me preguntas qué equipo elegir, tanto nuevo como de ocasión, te daría dos opciones:

Si quieres portabilidad: Macbook Pro, si lo vas a adquirir en el mercado de ocasión, compra uno con una antigüedad no superior a 2 años desde el año en que estés dispuesto a comprarlo.

Si por el contrario no tienes pensado moverte de un lado para otro, entonces te recomiendo un Mac mini. El Mac mini es la puerta de entrada a los ordenadores el desarrollo de Apps. Eso sí, necesitarás aparte un monitor, un teclado y un ratón; pues el Mac mini nuevo se vende sin estos accesorios, y en el mercado de ocasión suele venderse de la misma forma.

Por lo tanto, a nivel de hardware necesitaremos un/a ordenador/computadora Mac.

Antes también era necesario disponer de un dispositivo iOS, pero desde la versión 7 de Xcode, se pueden subir aplicaciones al App Store sin necesidad de tener un iPhone o similar.

## Software

A nivel de software, necesitamos un IDE (Integrated Development Environment) o dicho en castellano: una aplicación que nos proporcione las herramientas necesarias para crear Apps.

Todo IDE que se precie debería tener al menos un editor de código, herramientas para la creación de interfaces, un depurador para probar y ver posibles errores, así como un compilador.

Apple nos proporciona Xcode como aplicación integral para crear aplicaciones, lo podemos descargar de forma gratuita de la Mac App Store.

Veremos Xcode en profundidad en el capítulo 3 de este libro.

## ¿Qué es iOS?

---

iOS es el sistema operativo del iPhone, del iPad y del iPod Touch.

Estos son los únicos dispositivos de Apple que integran iOS como sistema operativo.

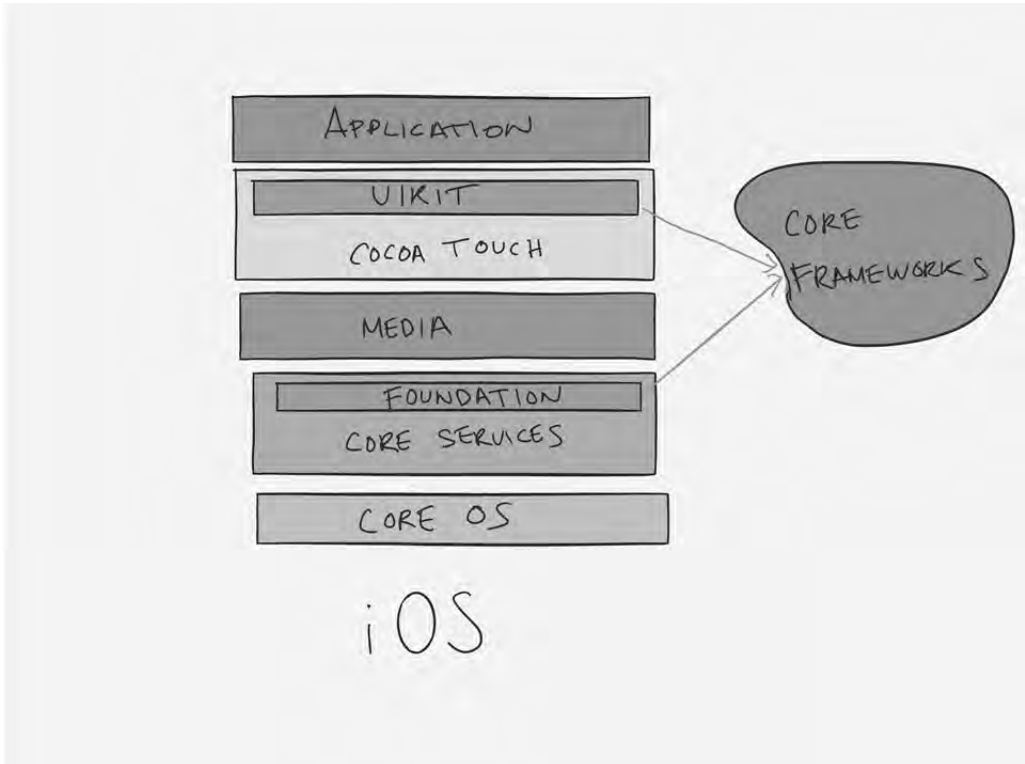
Los ordenadores Mac tienen el OS X como sistema operativo, el Apple Watch implementa watchOS, y el Apple TV, tvOS. Y aunque comparten algunos de sus Frameworks y patrones de diseño, son sistemas operativos diferentes.

El sistema operativo iOS fue presentado por Steve Jobs en la WWDC (World Wide Developer Conference) en el año 2007, llamado en sus primeras versiones iPhone OS, pues solamente el iPhone implementaba este sistema, no fue hasta el año 2010 y con su cuarta versión que tomó el nombre que tiene en la actualidad.

Actualmente está en su versión 11. Cada año Apple actualiza todos sus sistemas operativos, añadiendo nuevas funcionalidades para los usuarios, y lo más importante, añadiendo nuevas Clases y Frameworks para los desarrolladores.

iOS es un sistema operativo con base UNIX, a nivel de arquitectura cuenta con 4 capas como se ve en la figura siguiente:

- Touch
- Media
- Core Services
- Core OS



Dentro de Core Services se encuentra el Framework Foundation, y dentro de Cocoa Touch, UIKit.

Foundation y UIKit serán dos de los Frameworks que más usaremos.

Y ya hemos dicho que un Framework es un conjunto de clases, es decir, un gran almacén de objetos que podemos usar a nuestro antojo, por ejemplo un botón, una ventana, una cadena de caracteres, todo esto son objetos que ya están creados, y nosotros podemos acceder a estos Frameworks y usarlos en nuestras Apps.

## Tipos de Apps

---

A la hora de crear una aplicación podemos optar por diferentes tipos de arquitectura:

### App nativa con conectividad

Es la que se crea íntegramente con el entorno de desarrollo Xcode, la App puede funcionar sin acceso a internet, pues todo está compilado dentro de la aplicación. Su comercialización debe hacerse a través del App Store, y es necesario que pase por una revisión y posterior aprobación o rechazo por parte de Apple.

### Web App

Es una aplicación que se crea con formato web, se accede a ella a través de una dirección web, es decir, a través de una URL. En realidad es una web con el formato adecuado para su correcta visualización desde un dispositivo iOS. Se puede comercializar por nosotros mismos, sin necesidad de enviarla a Apple para su revisión. Para acceder a la App basta con acceder a la URL desde Safari.

### App Nativa + Web App

Se puede crear una App con el entorno de desarrollo de XCode, e implementar la Web dentro de la App nativa. UIKit nos provee de una clase llamada `UIWebView`, que lo que hace es mostrar una página web embebida en nuestra aplicación. Se comercializa en el App Store, previa aprobación por parte de Apple, la ventaja es que una vez aprobada, podemos actualizar su contenido sin necesidad de pasar por el proceso de aprobación de nuevo, pues el contenido dinámico se actualiza desde el servidor donde esté alojada la web.

### App nativa y Webservice

En este caso la aplicación se crearía con Xcode pero el modelo de datos estaría en un servidor. Hay transmisión de datos en uno u otro sentido. Es sin duda la estructura más utilizada.

## Algo más que programar

---

Espero que este capítulo te haya resultado interesante. Te advierto desde ya, que no todos los capítulos van a ser tan agradables.

Crear aplicaciones como otras tantas cosas, no es sencillo, pero también te digo que cualquier persona puede hacerlo.

Da igual la edad, da igual si estudiaste ciencias o letras, da igual si eras un auténtico fracaso en matemáticas, de verdad, hazme caso, todo eso da igual, no tienes que tener una carrera en Informática, o ser ingeniero para crear aplicaciones.

Todo es cuestión de actitud, en serio, si te lo propones puedes conseguirlo.

De hecho tú llevas programando desde que eras un niño, sí, créeme que programas desde niño, no de una forma consciente claro está, pero tu cerebro sigue la misma lógica que si estuvieses programando un algoritmo.

Parafraseando a mi colega de profesión Luis Berganza: *“programamos desde niños, recordáis cuando nuestra madre nos decía: o te comes la comida o no hay postre”*. Esto que nos decía nuestra madre, es una condición, si te comías la comida podías tomar postre, pero si no terminabas el plato, sencillamente te quedabas sin postre.

Llevando las palabras de nuestra madre al mundo de la programación, estaríamos hablando de una estructura de control, en programación usamos las estructuras de control para que nuestros programas, en nuestro caso Apps, tomen decisiones ellas mismas. En el caso que nos ocupa, la App sería el niño, y deberá tomar una decisión, tomar postre o no, dependiendo de si se come la comida.

En programación, las palabras que nos decía nuestra madre, se podrían traducir a algo como esto:

```
If ( te comes la comida ) {  
  
Puedes comerte el postre  
  
} else {  
  
No hay postre  
  
}
```

Lo que vemos arriba es una estructura de control que se usa muy a menudo en programación, en concreto la estructura de control **if-else**. Es común a la mayoría de lenguajes de programación, y en Swift también está disponible y la usaremos.

Veremos con más detalle esta y otras estructuras de control en el capítulo 5, cuando entremos de lleno a conocer Swift.

Yo no había programado en ningún lenguaje antes de aprender a crear aplicaciones, y conseguí aprender a programar, a crear bastantes Apps, subirlas al App Store y obtener ingresos con su venta, y tú puedes hacerlo lo mismo.

Conozco colegas de profesión que están ganando mucho dinero vendiendo sus aplicaciones, y no es que no hayan programado con anterioridad, no, es que no saben programar. Te preguntarás cómo pueden hacerlo entonces, pues muy sencillo, tienen fundamentos clave, conocen ciertos aspectos fundamentales de crear Apps, y luego buscan los recursos que necesitan en internet y saben cómo usarlos.

Parte de mi metodología consiste en eso, y lo veremos con más detenimiento en el capítulo 8. Te voy a dar ciertas pautas que son clave no solo que conozcas, sino que domines.

En tu camino de desarrollador de aplicaciones para móviles, podrás aprender a programar, incluso podrías llegar a ser un fantástico programador si tú quieres, pero si no quieres por el motivo que sea, también hay sitio para ti, porque crear aplicaciones consiste en algo más que programar.

Así pues no te preocupes si leyendo este libro, o incluso una vez terminado, no entiendes todas y cada una de las sentencias de un determinado algoritmo, no hace falta, vamos a usar programación orientada a objetos, y una de sus máximas es que no es importante saber cómo hace tal cosa un determinado objeto, sino qué puede hacer por ti ese objeto, cómo te puede ayudar a resolver tu problema el objeto en cuestión.

Tranquilo, he notado que fruncías el ceño cuando has leído la palabra algoritmo, es un término que da miedo, realmente da miedo, pero una vez conoces su significado la cosa cambia.

Un algoritmo no es más que una secuencia ordenada de instrucciones, una lista de acciones de lo que debe hacer tu aplicación en un momento dado.

Así, la programación es una parte de la creación de aplicaciones, pero no es la única, Apple cada vez facilita más las cosas para que tengamos que escribir menos código, e interactuemos más con los elementos que nos proporciona Xcode.

Por lo tanto, si no tienes conocimientos de programación, siempre puedes conocer una metodología, y después ir aprendiendo el lenguaje de programación poco a poco.

El símil es el siguiente, qué dirías si alguien te pregunta: ¿Puedo viajar a Londres sin saber inglés? ¡Pues claro que sí hombre!