



Apple
Coding
Academy



SWIFT DEVELOPER PROGRAM 2022



¿Quieres ser *developer* en entornos Apple pero no sabes por dónde empezar? En ese caso has llegado al lugar apropiado: **BIENVENIDO A SWIFT DEVELOPER PROGRAM 2022**, una formación intensiva en la que adquirirás las habilidades más importantes para comenzar tu andadura como desarrollador en entornos Apple.

Qué aprenderás

Ser desarrollador no es un camino fácil ni rápido. Desconfía de quien pretenda decirte que en unas semanas serás *developer*. No es cierto. En unas semanas, con una formación como este innovador programa sabrás:

- **Por dónde empezar.**
- **Hacia dónde dirigirte.** Sentarás las bases para saber cuál es la ruta que has de tomar.
- **Con qué cuentas.** Establecerás los fundamentos asentados en mejores prácticas, herramientas, arquitecturas, patrones, paradigmas, orientaciones, flujos...



Además, te mostrará el camino para:

- Convertirte en un *developer* de gran prestigio.
- Llegar a trabajar para una empresa (grande o pequeña) o establecerte por ti mismo y crear tu propio estudio de desarrollo.

Qué te ofrecemos

- **Un programa formativo único** y actualizado a las últimas versiones de iOS.
- **Formación progresiva** que te va guiando paso a paso de lo más simple a lo más complejo.
- **Clases en directo** basadas en trabajo real, como el que se haría en una consultora.
- **Grabación de las clases** para que puedas verlas cuanto necesites.
- **Ejercicios** resueltos en dinámicas grupales de trabajo.
- **Prácticas** propuestas para asentar tus conocimientos y ponerlos en práctica.
- **Proyecto tutorizado de fin de formación.**
- **Diploma acreditativo** con calificación en base a tu desempeño y esfuerzo.

Extras

- Acceso a la comunidad privada de AC Academy en Discord.
- 2 tutorías grupales de 90 minutos cada una.
- 1 tutoría *one to one* de 30 minutos con el formador.
- Complementos (Videos grabados).
 - ✓ Planificación de Proyectos.
 - ✓ Conoce Xcode.
 - ✓ Domina GIT.
 - ✓ Documentación con DooC.



Swift 5.7



Ser *developer* es hablar otro idioma: un nuevo lenguaje. Y en entornos Apple ese lenguaje se llama Swift. En **septiembre de 2022 Apple lanza Swift 5.7**, su última versión cargada de importantes novedades con la que podrás expresar los algoritmos que darán forma a tus desarrollos de manera práctica y eficiente.

Un lenguaje tiene su vocabulario, su gramática, su forma de ser construido y Swift no podía ser menos. Por eso, en este primer módulo del **SWIFT DEVELOPER PROGRAM 2022**, aprenderás a dominar Swift desde cero, tanto si conoces otros lenguajes de programación como si nunca has aprendido a programar. Incluso si crees que ya sabes Swift puede que realmente no lo conozcas como deberías.

Confía en nosotros porque **aprenderás a hablar y entender todo lo que expreses en lenguaje Swift** en vez de copiar y pegar código sin saber lo que haces, como el que usa un traductor online.



Temario

1. INTRODUCCIÓN AL DESARROLLO (MATERIAL GRABADO)

- Qué es desarrollo
- Datos, el pilar de cualquier programa
 - ✓ Mutabilidad
 - ✓ Datos binarios
 - ✓ Tipos de datos
 - ✓ Capacidad de un dato
- Algoritmos
 - ✓ Qué es un algoritmo
 - ✓ Divide y vencerás
 - ✓ Control de flujo y bucles
 - ✓ Diagramas de flujo
- Historia del desarrollo en entornos Apple

2. INICIACIÓN A SWIFT

- Tipos de datos
- Inferencia, datos vacíos y ámbito
- Operadores
- Flujos de control de código
- Datos opcionales
- Colecciones y tuplas
- Cadenas y expresiones regulares
- Casting de tipos de datos

3. PRÁCTICA DE EJERCICIOS BÁSICOS

4. ESTRUCTURAS EN SWIFT

- Funciones
- Programación orientada a objetos
- Clases
- Structs

- Enumeraciones
- Propiedades calculadas
- Observadores de propiedades
- Propiedades de instancia y estáticas
- Extensiones de tipos

5. PRÁCTICA DE EJERCICIOS INTERMEDIOS

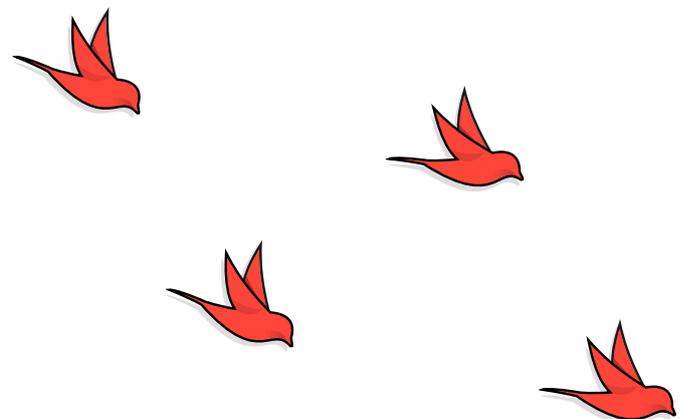
6. PATRONES DE DESARROLLO

- Control de errores
- Tipo de resultado
- Protocolos
- Patrón de delegados
- Patrón *singleton*
- Patrón de notificaciones
- Patrón de observadores
- Genéricos (Some y Any)

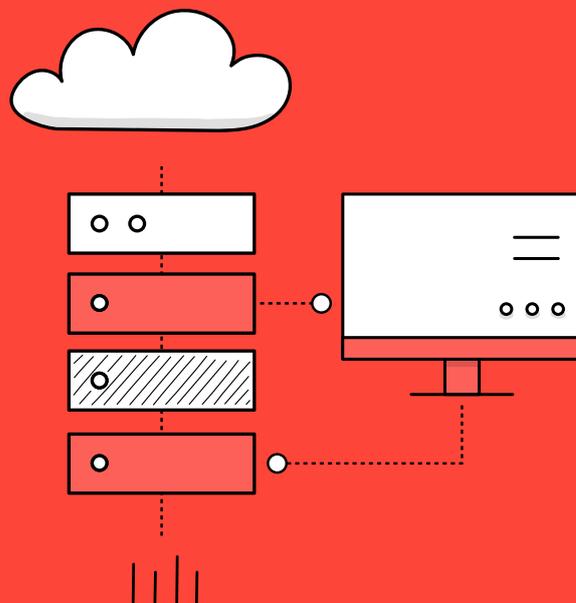
7. PRÁCTICA DE EJERCICIOS AVANZADOS

8. CODABLE, SERIALIZACIÓN DE DATOS

- Formato JSON
- Conversión de tipos implícitos
- Serialización de tipos complejos
- Serialización manual con control de errores



Concurrencia, asincronía y red con Async/Await



Hoy es imposible definir o entender una app sin hablar del uso de red para gestionar datos externamente a ella. El uso de red se hace en concurrencia —es decir, haciendo que varios procesos funcionen a la vez— lo que define el concepto de la asincronía de procesos.

Si no controlamos estos tres aspectos fundamentales (concurrencia, asincronía y red) en nuestro desarrollo, jamás estableceremos las bases necesarias para crecer como *developers* y utilizar las herramientas adecuadas en nuestros proyectos o en las empresas donde aspiramos a llegar.

Vamos a llevar tu código un paso más allá, dominando los procesos pesados de carga, de procesamiento o grabación de datos, manipulación individual o en bloque y/o el trabajo en red. Una parte esencial del conocimiento de cualquier *developer*.

Temario

1. TEORÍA

- Qué es concurrencia
- Componentes a usar: GCD, NSOperations y Async/Await
- Tipos de tareas
- Tipos de concurrencia
- Prioridad o calidad de servicio (QoS)
- Asincronía
- Data race
- Inversión de prioridad
- Bloqueo (*deadlock*)
- Buenas prácticas

2. TRABAJO BÁSICO EN RED

- URLSession
- URLRequest
- Patrón *callback*
- Patrón *callback* genérico
- Patrón *callback* avanzado con tipos de resultado y control de errores
- WebSocket

3. GRAND CENTRAL DISPATCH (GCD)

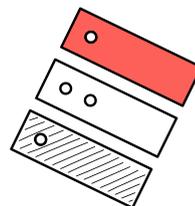
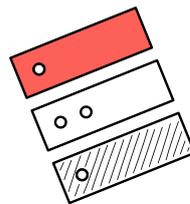
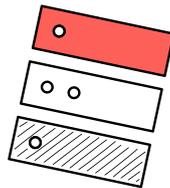
- Colas de despachado
- Grupos de trabajo
- Items de trabajo
- Barreras
- Data race en la práctica
- Semáforos
- Descarga por grupos y concurrente

4. ASYNC/AWAIT

- Del patrón *callback* hacia Async/Await.
- Tareas



- Continuaciones
- Concurrencia estructurada
- Grupos de tareas
- Streams asíncronos
- Secuencias asíncronas
- Data race
- Actores
- El protocolo Sendable
- Concurrencia distribuida



Programación de apps en iOS y iPadOS con SwiftUI



SwiftUI es la reinención del desarrollo de apps en entornos Apple. Usando este innovador *framework* codificas una vez y programas en todos los sistemas. Reinventa de cero el desarrollo en entornos Apple y lo lleva a cotas de sencillez, claridad y potencia nunca vistas en la plataforma.

En este módulo aprenderás a expresarte con SwiftUI sin librerías ni experimentos. Serás 100% nativo siguiendo las buenas prácticas de Apple de la WWDC 2022.

Temario

1. COMENZANDO CON SWIFTUI

- Qué es SwiftUI
- Evolución del *framework*
- Cómo funciona
- Componentes
- Modificadores
- Estructura básica de una vista
- San Francisco Symbols
- Xcode
- Catálogo de *assets*
- SwiftUI Preview (Canvas)

2. CREANDO INTERFACES CON SWIFTUI

- Componentes básicos
 - ✓ Texto
 - ✓ Imágenes
 - ✓ Formas
 - ✓ Espaciador dinámico
 - ✓ Botones
 - ✓ Etiquetas
 - ✓ Enlaces
 - ✓ Menús
 - ✓ Toolbars
- Modificadores
 - ✓ Estilos de fuentes y elementos
 - ✓ Tamaño del *frame*
 - ✓ Colores
 - ✓ Máscaras
 - ✓ Sombras
 - ✓ Capas (*background* y *overlay*)
 - ✓ Control de gestos

- Contenedores básicos
 - ✓ Vertical (*VStack*)
 - ✓ Horizontal (*HStack*)
 - ✓ Profundidad (*ZStack*)
 - ✓ Perezosos (*Lazy*)
 - ✓ Grupos de control y cajas
 - ✓ Formularios
 - ✓ Listas básicas
 - ✓ Secciones (cabeceras y pies)
 - ✓ Vistas de pestañas (*tabs*)
 - ✓ *Grid* Vertical
 - ✓ *Grid* Horizontal

3. ARQUITECTURA MVVM Y LOS DATOS

- Qué es Modelo Vista Vista/Modelo
- Diagrama y ciclo de vida de una app SwiftUI
- Empaquetadores de propiedad (*Property Wrappers*)
- El estado (*State*)
- Enlaces (*Bindings*)
- Objetos Observables (*ObservableObjects*)
- Propiedades publicadas (*Published*)
- Objeto observado (*ObservedObject*)
- ViewModel Global u Objetos de Estado (*StateObject*)
- Objetos de entorno (*EnvironmentObject*)
- Variables de entorno

4. CONSTRUYENDO APPS CON DATOS

- Navegación y dependencias
 - ✓ Listas
 - ✓ Selección de elementos
 - ✓ Selección múltiple
 - ✓ Stack de navegación
 - ✓ Navegación basada en datos
 - ✓ Navegación retrocompatible
 - ✓ Máquinas de estado



- ✓ Transiciones entre estados
- ✓ Animando elementos entre distintos estados
- Componentes interactivos
 - ✓ Selectores de datos
 - ✓ Selectores de fechas
 - ✓ Campos de texto
 - ✓ Campos multilínea
 - ✓ Editores de texto
 - ✓ Grupos desplegados y definidos (*Disclosure/Outline*)
 - ✓ Vistas de progreso y *sliders*
 - ✓ Switches (en estilo interruptor o botón)
 - ✓ Stepper
- Formularios
 - ✓ Campos de texto
 - ✓ Control de foco
 - ✓ Tipos de teclado
 - ✓ Campos de texto multilínea
 - ✓ Control de errores en formularios
 - ✓ Alertas
 - ✓ Edición de campos *inline*
 - ✓ Flujos entre varios *ViewModels*

5. DATOS EN LA NUBE

- Tareas asíncronas
- Flujo de trabajo asíncrono
- Interfaz de red
- Gestión de tareas pesadas

6. SWIFTUI MÁS ALLÁ

- Contenedores complejos
 - ✓ *Grid* genérico (filas y columnas)
 - ✓ Vista de *scroll* y lector
 - ✓ Lector de geometría (*GeometryReader*)
 - ✓ Claves de preferencia

- ✓ Vistas que encajan (*ViewThatFits*)
- ✓ El protocolo *Layout*
- ✓ Claves de valores de *layout*
- ✓ Tipo «Cualquier *Layout*» (*AnyLayout*)
- ✓ Canvas
- Modularización
 - ✓ Estilos propios de vistas
 - ✓ Estilos de botones
 - ✓ Creación de componentes propios
 - ✓ Componentes de UIKit
- SwiftUI en el iPad
 - ✓ Tablas multi-columnas
 - ✓ Selección de datos y menús contextuales
 - ✓ Split View de 2 columnas
 - ✓ Split View de 3 columnas
 - ✓ Toolbars personalizadas
 - ✓ Menús desplegados de acciones
 - ✓ Documentos asociados a la navegación
- Importando y exportando datos
 - ✓ El protocolo Transferable
 - ✓ Compartir contenido
 - ✓ Acceso a la biblioteca de medios
 - ✓ Drag & Drop
 - ✓ Cargando y grabando ficheros en Files
 - ✓ Asociando tipos de archivo a nuestra app
- Widgets
 - ✓ Widgets estáticos
 - ✓ Widgets dinámicos
 - ✓ Complicaciones en *WidgetKit* para la pantalla de bloqueo
 - ✓ Actividades en vivo (*Live Activities*)



7. SWIFT CHARTS

- Modelo de datos para gráficas
- Tipos de Marks
- Información integrada en la gráfica
- Animación de resultados
- Descripciones y detalles
- Gráficas circulares

8. FLUJO BÁSICO DE TESTING Y TDD

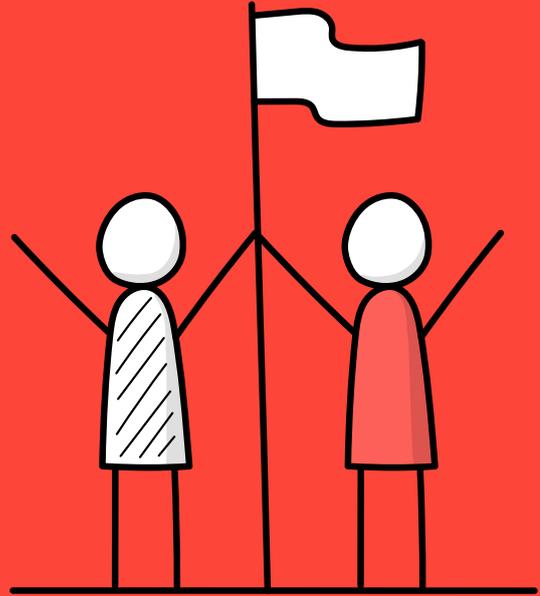
- Definición del modelo
- Persistencia del modelo
- Persistencia de red
- Modelo lógico
- Vista Modelo local
- Vista Modelo global

Aunque el programa de SwiftUI mantiene un orden para exponer el temario, este no será respetado en el transcurso de la formación para maximizar el entendimiento de los distintos conceptos y cómo se interrelacionan interfaz, arquitectura, testing, componentes y demás elementos.

SwiftUI será mostrado de forma modular y se explorarán todas sus posibilidades, pero las partes más abstractas, los ejemplos prácticos, la arquitectura y el testing, se expondrán mediante el desarrollo de ciertas apps de prueba con funcionalidades y flujos destinados a ejemplificar de una forma clara la interacción entre las distintas partes y elementos para construir un todo.

Se harán pantallas únicas, mini-apps y conceptos concretos, así como apps más grandes tanto de persistencia local como en red. Explotando todas las opciones de APIs propias y externas que permitan poner en práctica todo lo aprendido. A esto se unen dos proyectos que se propondrán a los alumnos para su resolución y que evaluarán la formación en su conjunto.

Proyecto final



Al final del programa se propondrá un proyecto concreto de construcción de una app en SwiftUI para consumir una API REST creada por Apple Coding Academy, con el propósito de poner en práctica todos los conocimientos adquiridos por el alumno.

Dicho proyecto permitirá aplicar los conocimientos de Swift, concurrencia, red, asincronía y las partes fundamentales de SwiftUI. Habrá distintos retos para llevar la app a diferentes metas que supondrán distintas calificaciones en función de cuánto se avance en la misma y su acabado.

El nivel de implicación y la calidad de tu trabajo determinarán la calificación final.

Información general



¿Estás preparado para iniciar tu camino como *developer*? Si es así, aquí podrás consultar los datos relativos a la fecha de impartición, modalidad de la formación, duración o precios.



Duración

128 horas, 32 días lectivos
5 semanas de proyecto



Calendario lectivo y tutorías

Del 4 de octubre al 30 de noviembre*

*Los días 12 de octubre y 1 de noviembre no hay clase

Tutorías: 29 de octubre y 19 de noviembre

OCTUBRE						
L	M	X	J	V	S	D
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

NOVIEMBRE						
L	M	X	J	V	S	D
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				



**Proyecto**

Del 1 de diciembre de 2022 al 5 de enero de 2023

**Horario**

De lunes a jueves de 19:00 a 23:00 horas
(hora española)

**Modalidad**

Online en directo

**Formador**

Julio César Fernández Muñoz

**Precio de la formación**

3.195€

**Reserva**

300€ que se descontarán
del precio total al formalizar la matrícula

**Descuentos**

- 10% para antiguos alumnos
- 4% por reserva antes del 5 de septiembre

**Formas de pago**

- 1 pago único
- 2 pagos
 - ✓ 50% el 1/10/2022 (descontando la reserva)
 - ✓ 50% el 1/11/2022
- Otras formas de pago: consultar

**Formación bonificada por FUNDAE**

para las empresas que quieran inscribir a algún trabajador

**Información y contratación**

hola@acoding.academy

Entrevista personal

acoding.academy





Anabel Segura, 10, planta 3
28108, Alcobendas (Madrid)



acoding.academy



+34 91 184 64 22



hola@acoding.academy