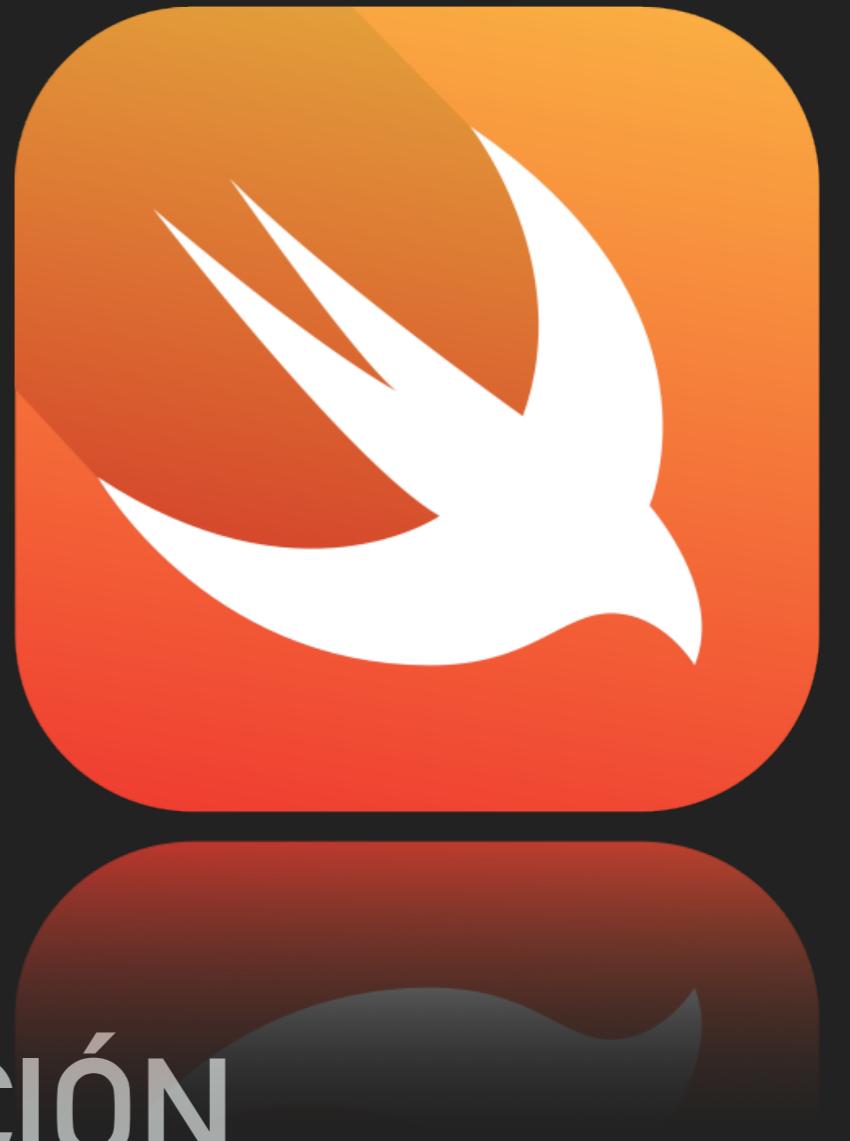


# SWIFT

---



SWIFTS ARE THE FASTEST OF BIRDS. LARGER SPECIES ARE AMONGST THE FASTEST FLIERS IN THE ANIMAL KINGDOM, WITH THE **WHITE-THROATED NEEDLETAIL** HAVING BEEN REPORTED FLYING AT UP TO 169 KM/H (105 MPH). EVEN THE **COMMON SWIFT** CAN CRUISE AT A MAXIMUM SPEED OF 31 **METRES PER SECOND** (112 KM/H; 70 MPH). IN A SINGLE YEAR THE COMMON SWIFT CAN COVER AT LEAST 200,000 KM. [HTTPS://EN.WIKIPEDIA.ORG/WIKI/SWIFT](https://en.wikipedia.org/wiki/Swift)



LENGUAJE DE PROGRAMACIÓN

---

**SWIFT**

Swift is a general-purpose programming language built using a modern approach to safety, performance, and software design patterns.

The goal of the Swift project is to create the best available language for uses ranging from systems programming, to mobile and desktop apps, scaling up to cloud services. Most importantly, Swift is designed to make writing and maintaining *correct* programs easier for the developer.

swift programming language creator: Chris Lattner

## WORK WITH

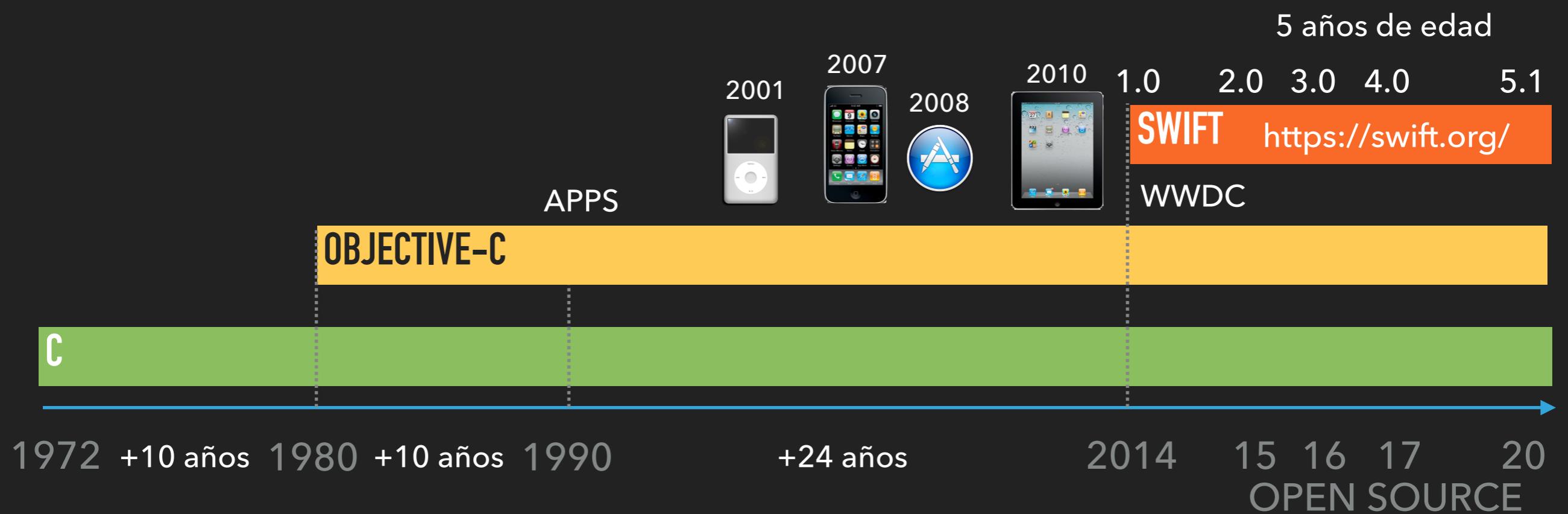
COCOA IS APPLE'S NATIVE OBJECT-ORIENTED APPLICATION PROGRAMMING INTERFACE (API) FOR THEIR OPERATING SYSTEM MACOS

COCOA CONSISTS OF THE FOUNDATION KIT, APPLICATION KIT, AND CORE DATA FRAMEWORKS, AS INCLUDED BY THE **COCOA.H** HEADER FILE, AND THE LIBRARIES AND FRAMEWORKS INCLUDED

BY THOSE, SUCH AS THE C STANDARD LIBRARY AND THE OBJECTIVE-C RUNTIME

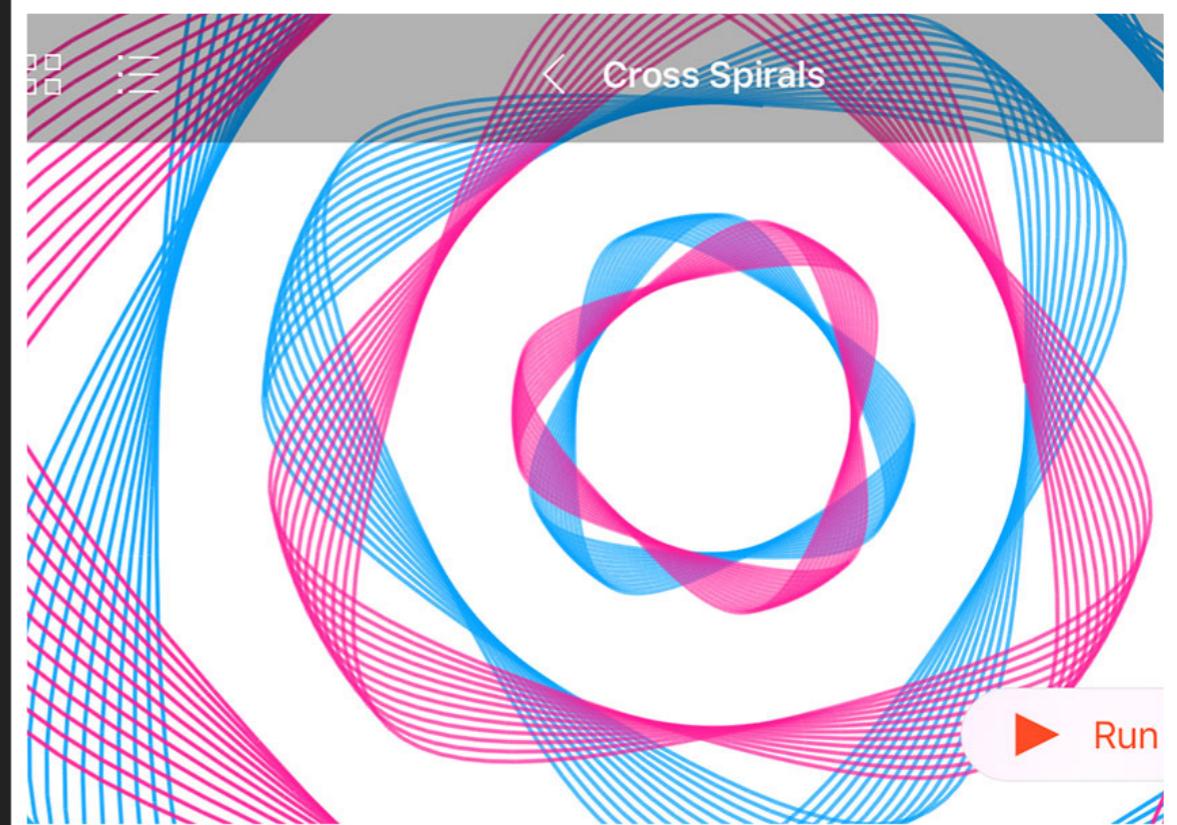
FOR **IOS, TVOS, AND WATCHOS**, A SIMILAR API EXISTS, NAMED COCOA TOUCH, WHICH INCLUDES GESTURE RECOGNITION, ANIMATION, AND A DIFFERENT SET OF GRAPHICAL CONTROL ELEMENTS. IT IS USED IN APPLICATIONS FOR APPLE DEVICES SUCH AS IPHONE, IPAD, IPOD TOUCH, APPLE TV, AND APPLE WATCH.

# HISTORIA



# UN LENGUAJE MODERNO

- ▶ OBJETIVO: SEGURO, RAPIDO Y CLARO
- ▶ CARACTERÍSTICAS
  - ▶ OPEN SOURCE
  - ▶ SINTAXIS CLARA
  - ▶ INFERENCIA DE TIPOS
  - ▶ ESCRIBIR CÓDIGO CON SEGURIDAD
  - ▶ CONTEO DE REFERENCIA AUTOMÁTICO
  - ▶ TUPLAS Y MÚLTIPLES VALORES DE RETORNO
  - ▶ GENÉRICOS



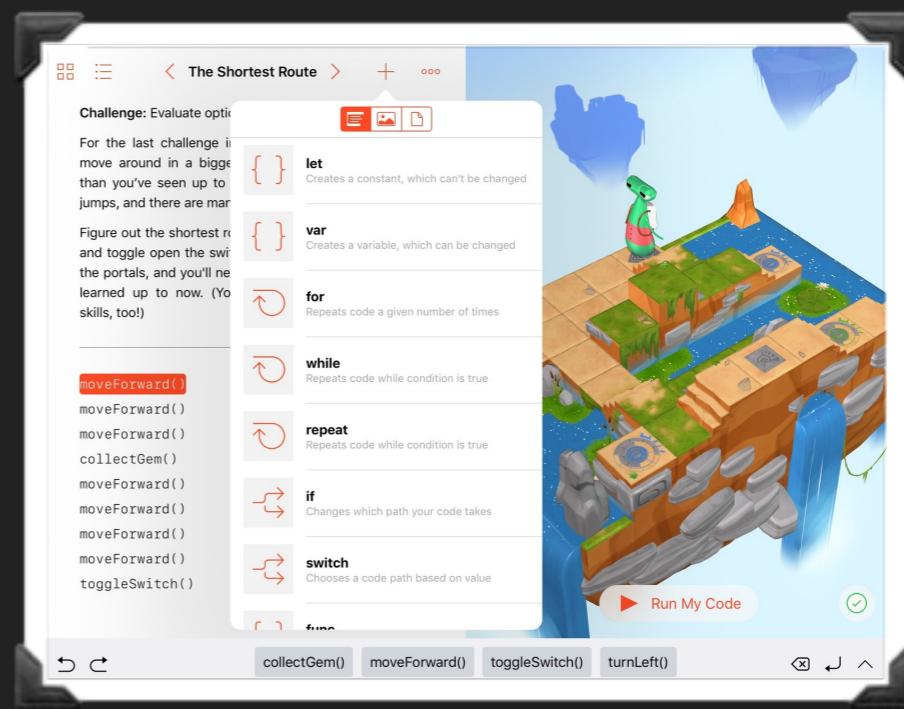
```
var spirals: [Spiral] = []  
  
// Spiral Colors  
let colorPrimary = UIColor(hue: 0.561, saturation:  
    0.973, brightness: 0.997, alpha: 0.600)  
let colorSecondary = UIColor(hue: 0.909, saturation:  
    0.878, brightness: 1.000, alpha: 0.600)  
  
// Configuration for spiral  
struct SpiralConfig {  
    let alpha: Double  
    let scale: Double
```



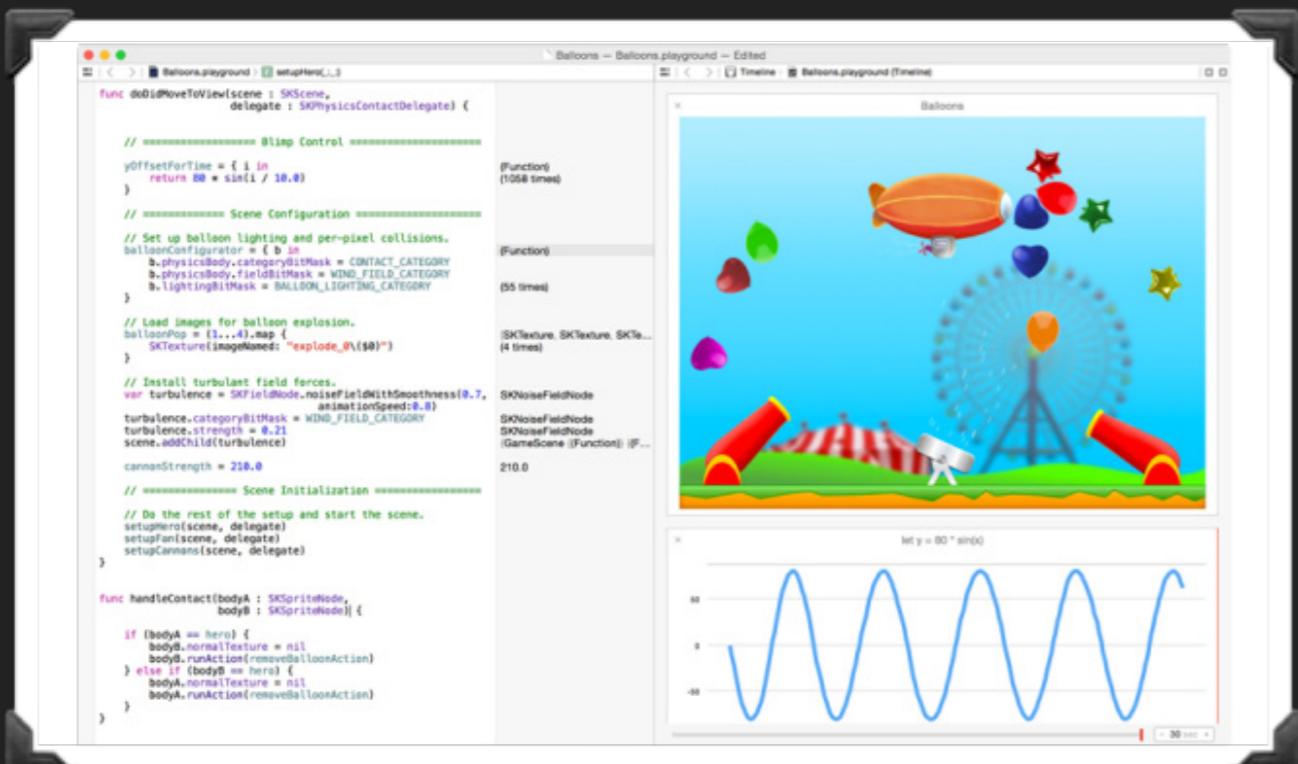
## UN LENGUAJE MODERNO

- ▶ ITERACIÓN FÁCIL SOBRE COLECCIONES
- ▶ ESTRUCTURAS CON METODOS, EXTENSIONES Y PROTOCOLOS
- ▶ PATRONES FUNCIONALES. EJ. MAP, FILTROS, REDUCIR
- ▶ MANEJO DE ERRORES Y DE CONTROL DE FLUJO AVANZADO
- ▶ OPCIONALES
- ▶ CLOSURES

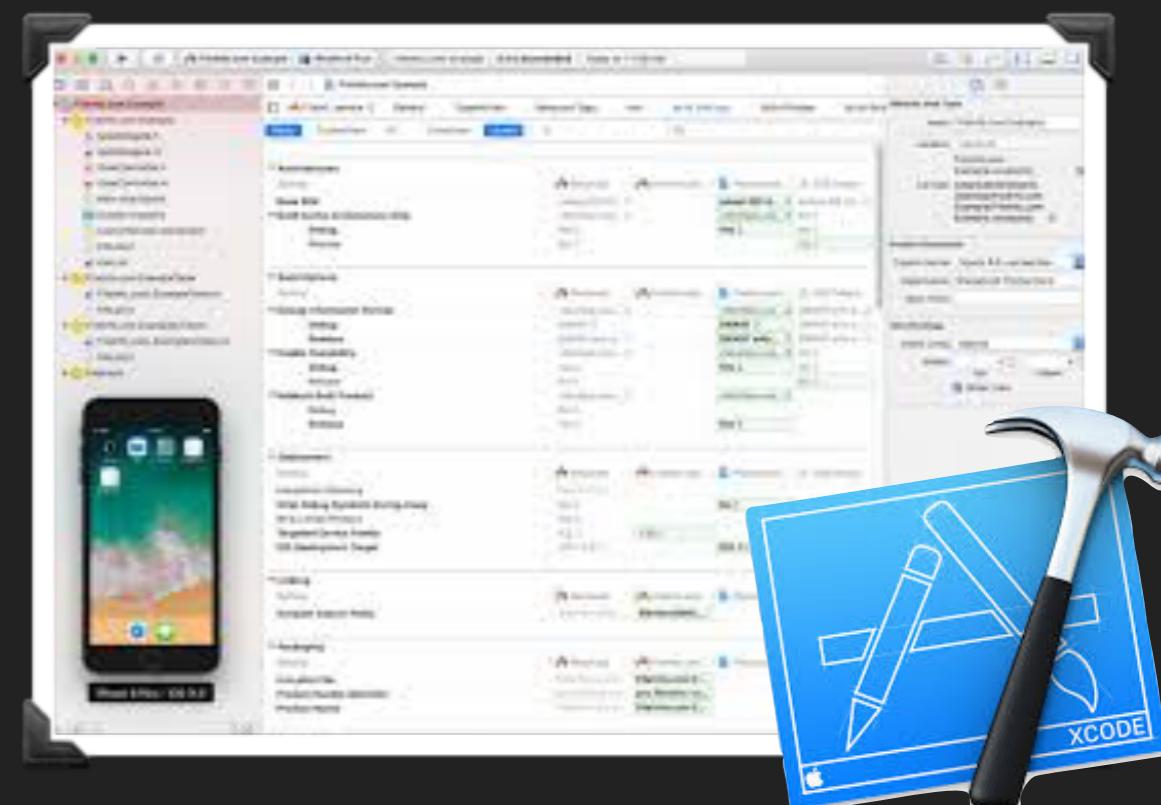
# ENTORNOS DE DESARROLLO



Swift Playground



Playground



XCode 9.X

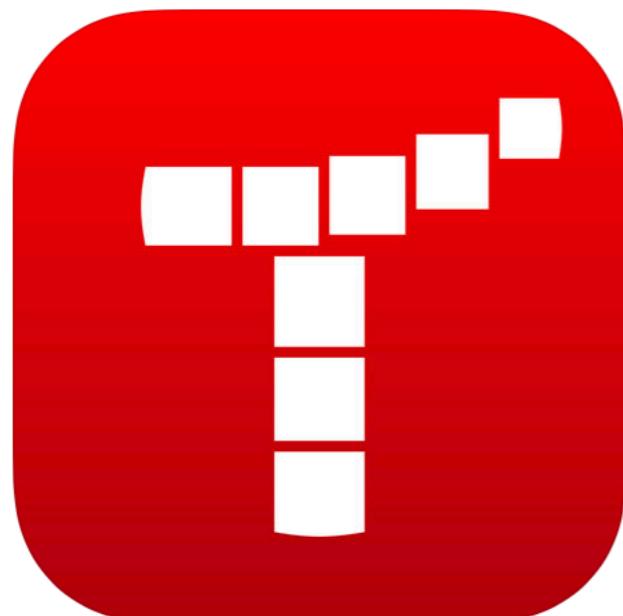
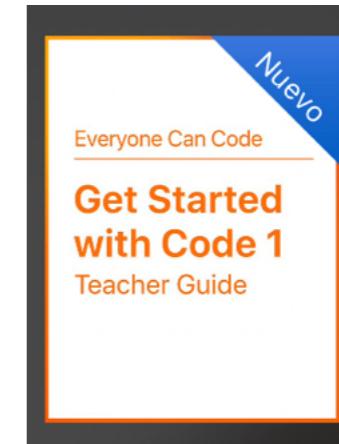


## codeSpark Academy 4+

Kids Coding with The Foos  
codeSpark

★★★★★ 1.3K Ratings

Free • Offers In-App Purchases

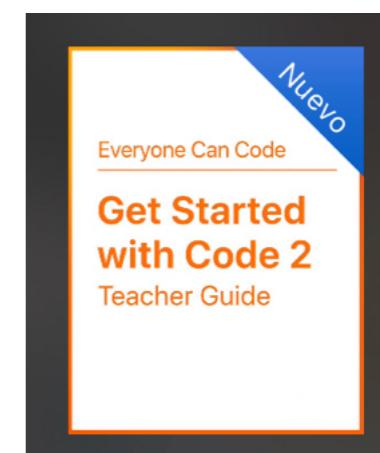


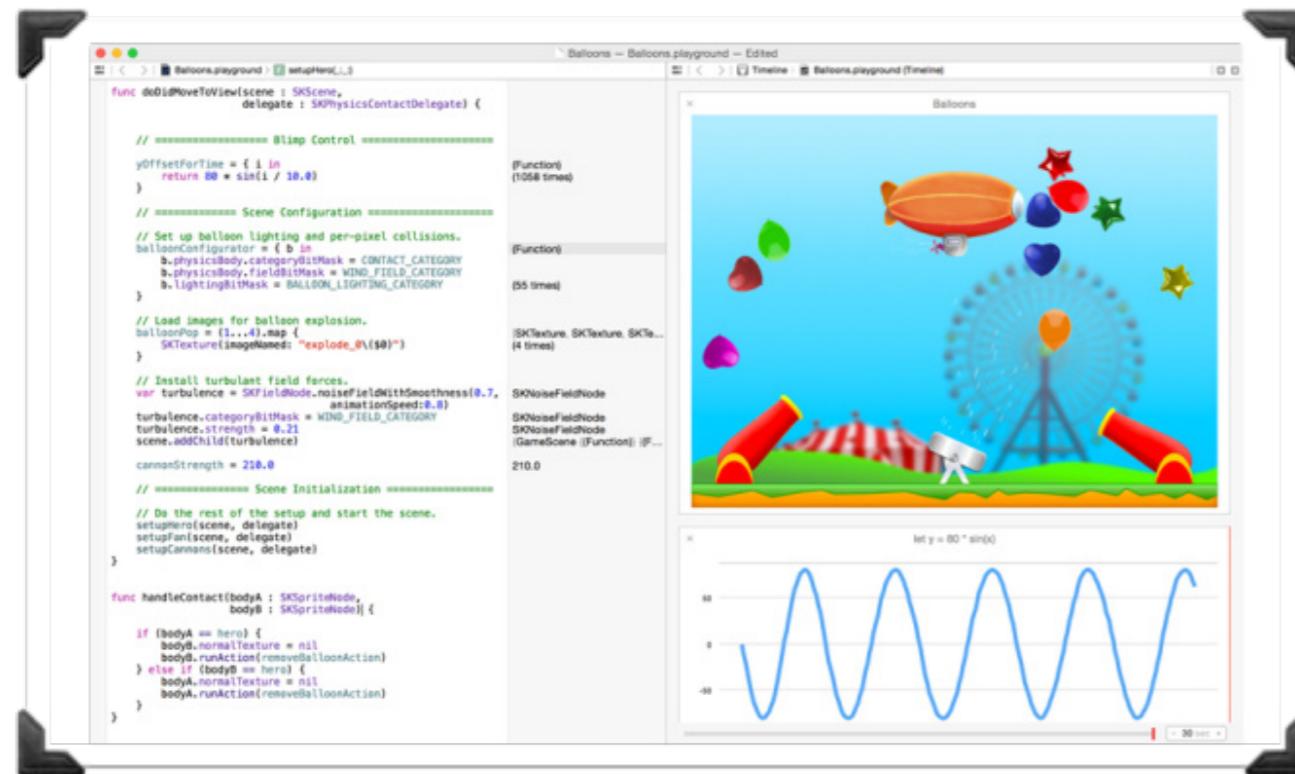
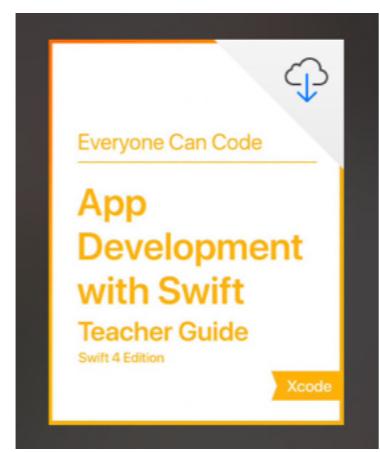
## Tynker: Coding for Kids 4+

Coding and Minecraft Modding  
Tynker

★★★★★ 1.3K Ratings

Free • Offers In-App Purchases





Balloons -- Balloons.playground -- Edited

```
func didMoveToView(scene : SKScene, delegate : SKPhysicsContactDelegate) {
    // Blimp Control
    yOffsetForTime = { l in
        return 80 * sin(l / 18.0)
    }

    // Scene Configuration
    // Set up balloon lighting and per-pixel collisions.
    balloonConfigurator = { b in
        b.physicsBody.categoryBitMask = CONTACT_CATEGORY
        b.physicsBody.fieldBitMask = WIND_FIELD_CATEGORY
        b.lightingBitMask = BALLOON_LIGHTING_CATEGORY
    }

    // Load images for balloon explosion.
    balloonPop = [1...4].map {
        SKTexture(imageNamed: "explode_0\(\($0)\)")
    }

    // Install turbulent field forces.
    var turbulence = SKFieldNode.noiseFieldWithSmoothness(0.7,
        animationSpeed: 0.8)
    turbulence.categoryBitMask = WIND_FIELD_CATEGORY
    turbulence.strength = 0.21
    scene.addChild(turbulence)

    cannonStrength = 210.0
}

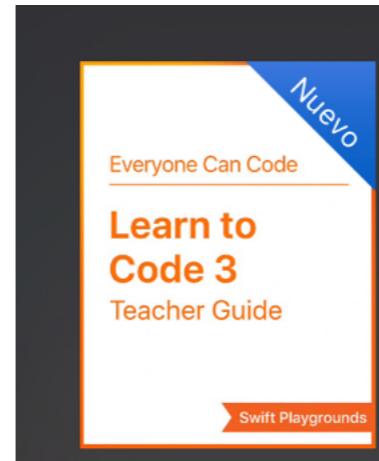
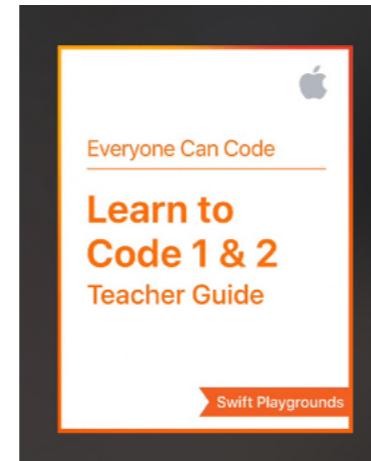
// Scene Initialization
// Do the rest of the setup and start the scene.
setupHeroicScene(delegate)
setupPanisScene(delegate)
setupCannonsScene(delegate)

func handleContact(bodyA : SKSpriteNode, bodyB : SKSpriteNode) {
    if (bodyA == hero) {
        bodyA.normalTexture = nil
        bodyA.runAction(removeBalloonAction)
    } else if (bodyB == hero) {
        bodyB.normalTexture = nil
        bodyA.runAction(removeBalloonAction)
    }
}
```

The playground shows a game scene with a blimp, balloons, and cannons. Below the scene is a graph of a sine wave labeled `let y = 80 * sin(x)`.



## SWIFT PLAYGROUND



Cancelar

Agregar suscripción    Editar



Aprende a programar 1  
Swift 4 Edition



Aprende a programar 2  
Swift 4 Edition



Aprende a programar 3  
Swift 4 Edition

OBTENER

OBTENER

OBTENER

### Retos



Máquina de código  
Principiante  
Swift 4 Edition

OBTENER



El mensaje cifrado  
(parte 1)  
Intermedio  
Swift 4 Edition

OBTENER



Piedra, papel  
o tijeras  
Principiante  
Swift 4 Edition

OBTENER



Hora del código  
Principiante  
Swift 4 Edition

OBTENER



Batalla naval  
Intermedio  
Swift 4 Edition

OBTENER



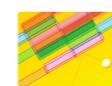
Escape del laberinto  
Intermedio  
Swift 4 Edition

OBTENER



Realidad aumentada  
Intermedio  
Swift 4 Edition

OBTENER



Brick Breaker  
Intermedio  
Swift 4 Edition

OBTENER



Espirales  
Principiante  
Swift 4 Edition

OBTENER

### Puntos de partida



En blanco  
Swift 4 Edition

OBTENER



Respuestas  
Swift 4 Edition

OBTENER



Mundo de  
rompecabezas  
Swift 4 Edition

OBTENER

# MATERIAL DE APOYO DISPONIBLE

## IBOOKS

### Everyone Can Code

Apple Education

